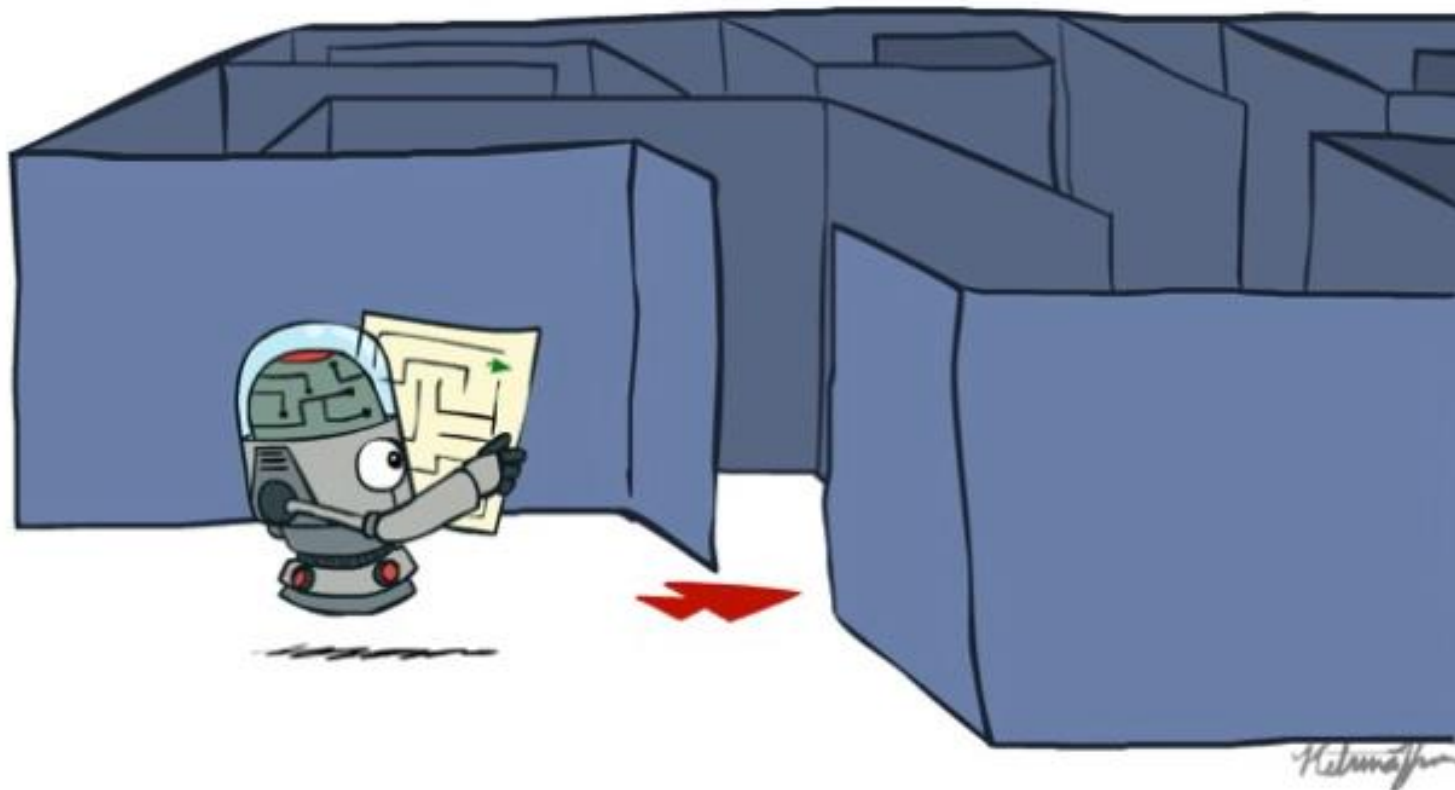


جستجوی ناآگاهانه

سید ناصر رضوی n.razavi@tabrizu.ac.ir

۱۳۹۷



یادآوری: مسائل جستجو

۳

□ یک مسئله جستجو شامل موارد زیر است:

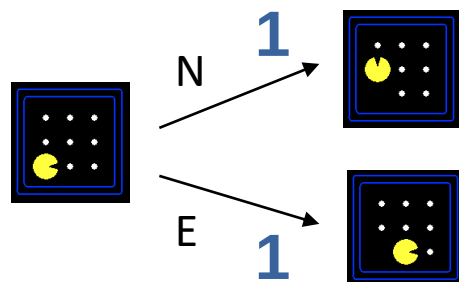
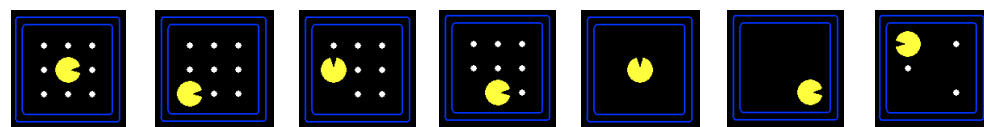
□ یک فضای حالت

□ یک تابع جانشین:

■ بیانگر عملیات قابل انجام در هر حالت و هزینه و نتیجه هر عمل

■ بیانگر چگونگی تغییر محیط در پاسخ به اعمال عامل

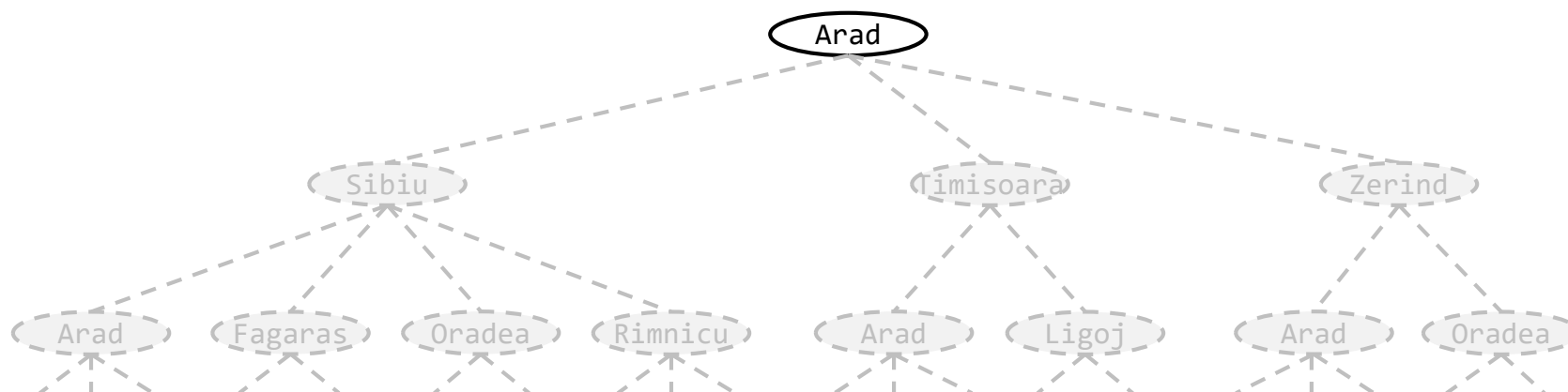
□ یک حالت شروع و یک حالت هدف



□ راه حل. دنباله‌ای از عملیات که حالت شروع را به یک حالت هدف تبدیل می‌کند.

یادآوری: جستجوی فضای حالت با استفاده از درخت جستجو

۴



□ جستجو.

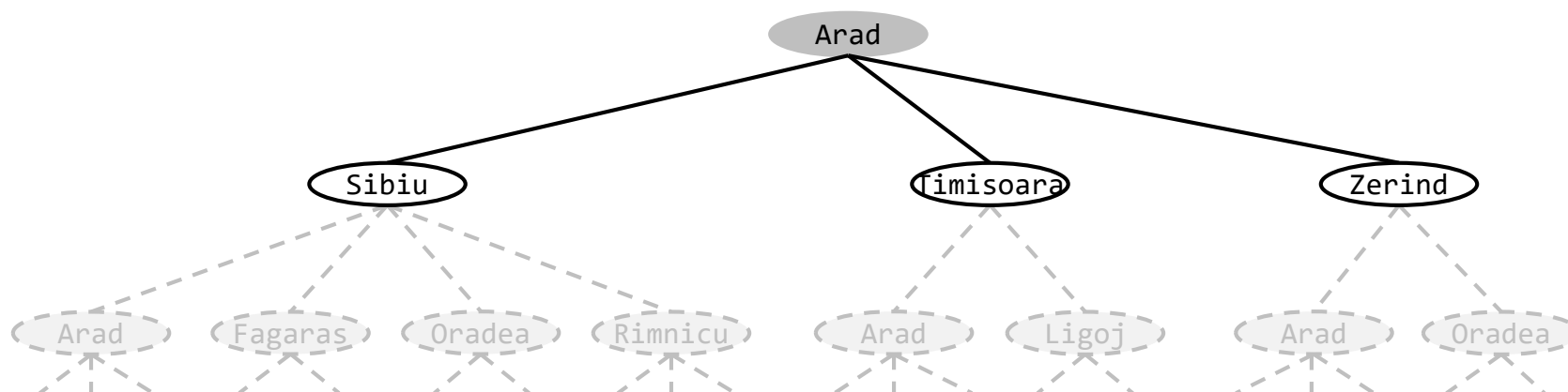
□ انتخاب یک گره نامزد

□ آزمایش هدف

□ گسترش گره انتخاب شده

یادآوری: جستجوی فضای حالت با استفاده از درخت جستجو

۵



□ جستجو.

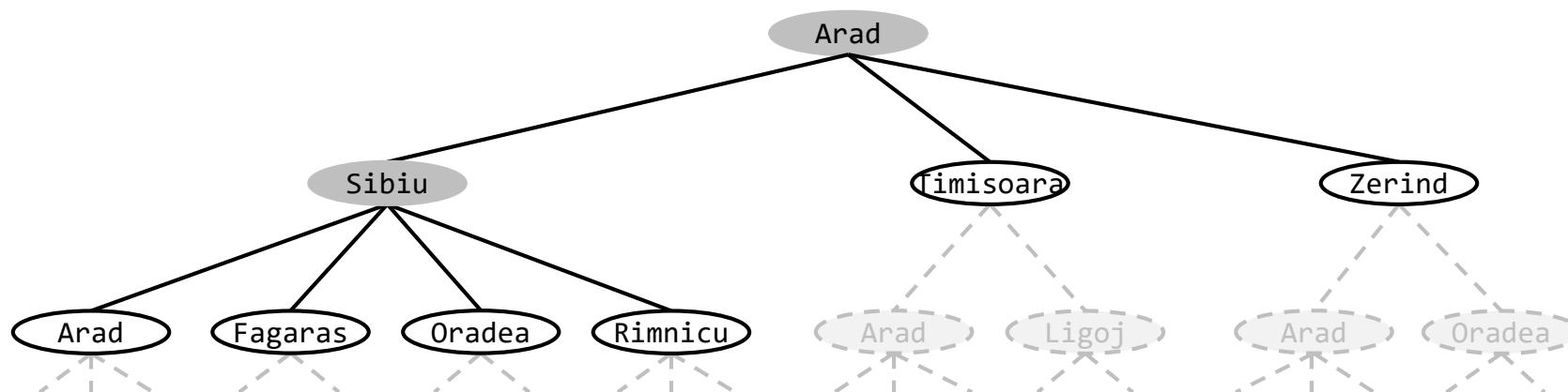
□ انتخاب یک گره نامزد

□ آزمایش هدف

□ گسترش گره انتخاب شده

یادآوری: جستجوی فضای حالت با استفاده از درخت جستجو

۶



□ جستجو.

□ انتخاب یک گره نامزد

□ آزمایش هدف

□ گسترش گره انتخاب شده

یادآوری: جستجوی درختی

۷

```
function TREE-SEARCH(problem) returns a solution, or failure
  initialize the frontier using the initial state of problem
  loop do
    if the frontier is empty then return failure
    choose a leaf node and remove it from the frontier
    if the node contains a goal state then return the corresponding solution
    expand the chosen node, adding the resulting nodes to the frontier
  end
```

□ ایده‌های مهم.

□ استراتژی گسترش دادن

□ ساختمان داده fringe

□ گسترش دادن گره‌ها

هر بار کدام گره درخت باید گسترش داده شود؟

استراتژی‌های ناآگاهانه

۸

□ استراتژی ناآگاهانه (کور).

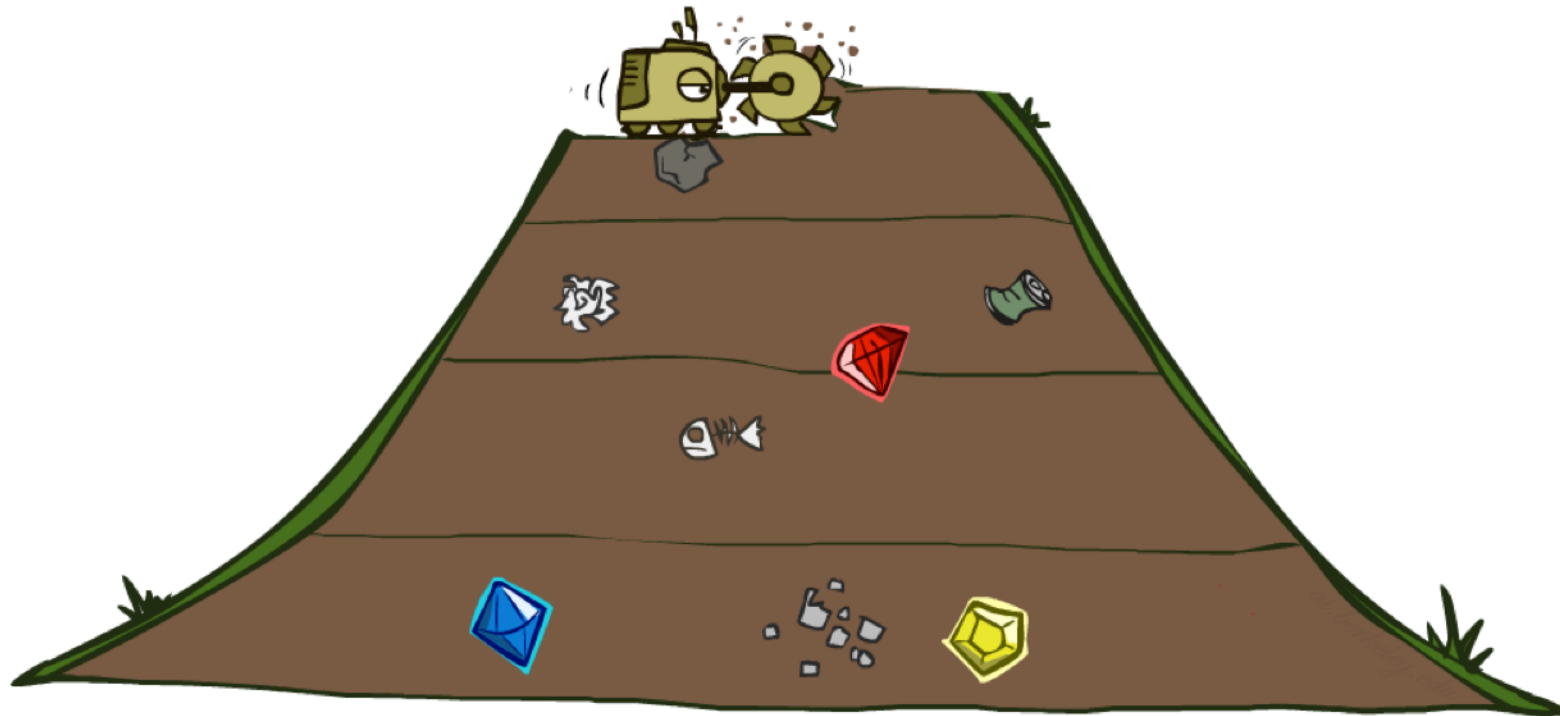
- تنها از اطلاعات موجود در تعریف مسئله برای جستجو استفاده می‌کند.
- تنها می‌تواند حالت‌های هدف را از حالت‌های غیر هدف تشخیص دهد.

□ انواع استراتژی‌های ناآگاهانه.

- جستجوی سطحی
- جستجوی هزینه یکنواخت
- جستجوی عمقی
- جستجوی عمقی محدود
- جستجوی عمیق کننده تکراری

جستجوی سطحی

۹

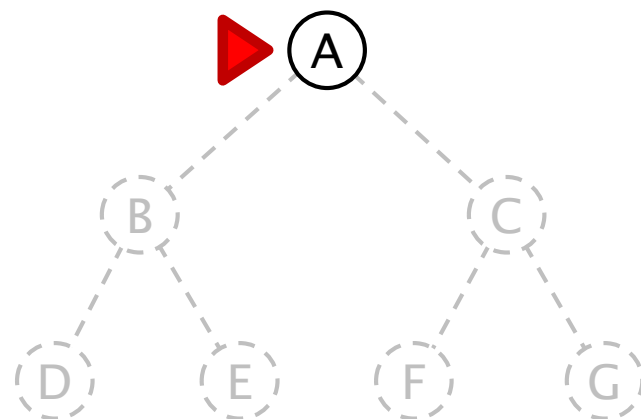


جستجوی سطحی

۱۰

□ جستجوی سطحی.

□ هر بار **سطحی ترین** گره گسترش نیافته را گسترش می دهد.



□ پیاده سازی.

□ fringe یک صف FIFO است.

□ فرزندان جدید به انتهای صف اضافه می شوند.

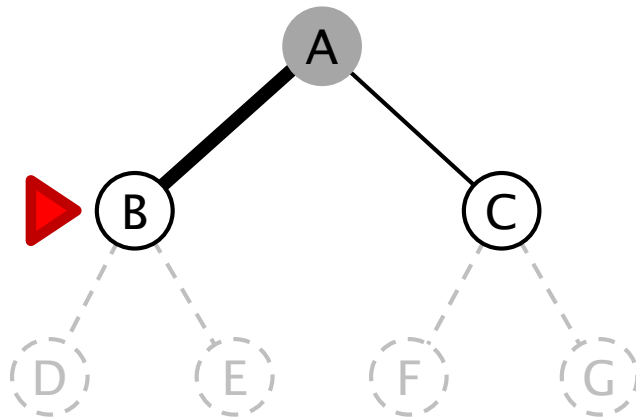
fringe: [A]

جستجوی سطحی

۱۱

□ جستجوی سطحی.

□ هر بار **سطحی ترین** گره گسترش نیافته را گسترش می دهد.



□ پیاده سازی.

□ fringe یک صف FIFO است.

□ فرزندان جدید به انتهای صف اضافه می شوند.

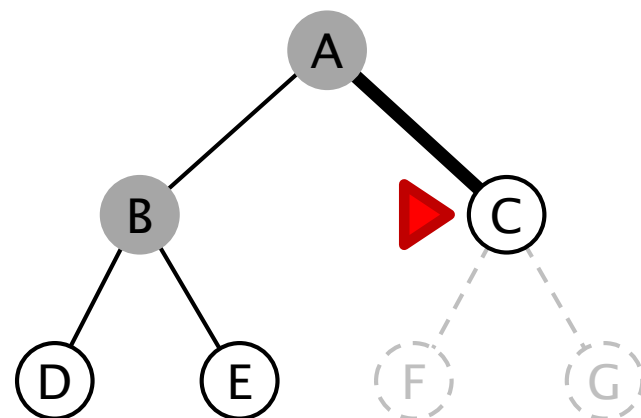
fringe: [B, C]

جستجوی سطحی

۱۲

□ جستجوی سطحی.

□ هر بار **سطحی ترین** گره گسترش نیافته را گسترش می دهد.



□ پیاده سازی.

□ fringe یک صف FIFO است.

□ فرزندان جدید به انتهای صف اضافه می شوند.

fringe: [C, D, E]

جستجوی سطحی

۱۳

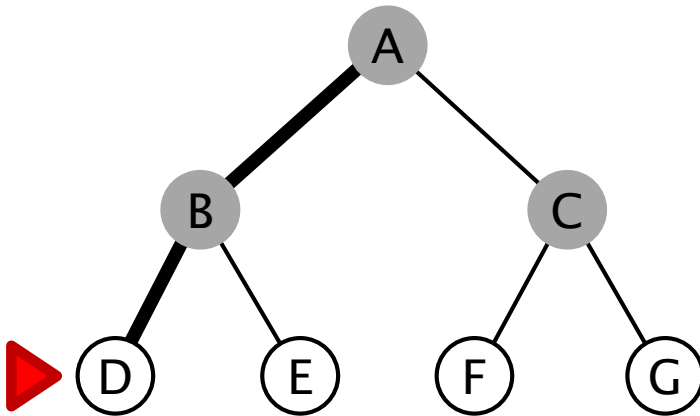
□ جستجوی سطحی.

□ هر بار **سطحی ترین** گره گسترش نیافته را گسترش می دهد.

□ پیاده سازی.

□ fringe یک صف FIFO است.

□ فرزندان جدید به انتهای صف اضافه می شوند.



fringe: [D, E, F, G]

```
function BREADTH-FIRST-SEARCH(problem) returns a solution, or failure
  node  $\leftarrow$  a node with STATE = problem.INITIAL-STATE, PATH-COST = 0
  if problem.GOAL-TEST(node.STATE) then return SOLUTION(node)
  frontier  $\leftarrow$  a FIFO queue with node as the only element
  explored  $\leftarrow$  an empty set
  loop do
    if EMPTY?(frontier) then return failure
    node  $\leftarrow$  POP(frontier) /* chooses the shallowest node in frontier */
    add node.STATE to explored
    for each action in problem.ACTIONS(node.STATE) do
      child  $\leftarrow$  CHILD-NODE(problem, node, action)
      if child.STATE not in explored or frontier then
        if problem.GOAL-TEST(child.STATE) then return SOLUTION(child)
        frontier  $\leftarrow$  INSERT(child, frontier)
```

□ توجه. آزمایش هدف در لحظه تولید گره‌ها انجام می‌شود و نه در لحظه گسترش.

یادآوری: ارزیابی استراتژی‌های جستجو

۱۵

□ استراتژی جستجو. ترتیب **گسترش دادن** گره‌ها.

□ ارزیابی یک استراتژی.

- **کامل بودن**: آیا الگوریتم در صورت وجود راه‌حل، یافتن راه‌حل را تضمین می‌کند؟
- **بهینه بودن**: آیا الگوریتم همواره کم‌هزینه‌ترین راه‌حل (راه‌حل بهینه) را پیدا می‌کند؟
- **پیچیدگی زمانی**: یافتن راه‌حل به چه میزان زمان نیاز دارد؟
- **پیچیدگی حافظه**: برای انجام جستجو به چه میزان حافظه نیاز است؟

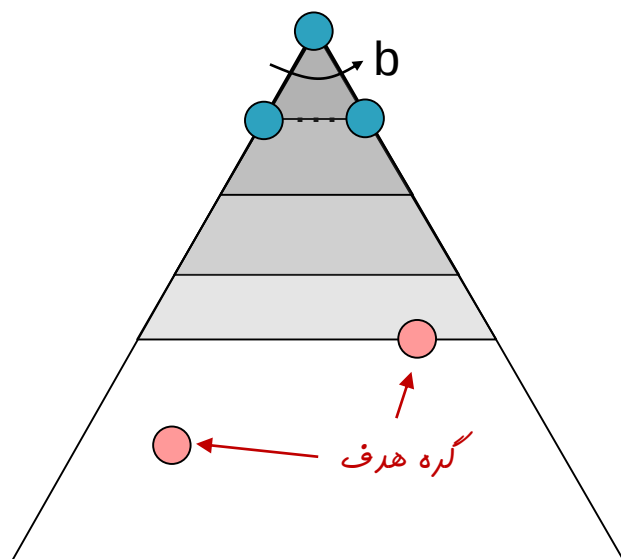
□ پارامترهای مهم.

- **b**: حداکثر تعداد فرزندان یک گره در درخت جستجو
- **d**: عمق کم‌عمق‌ترین گره هدف در درخت جستجو
- **m**: حداکثر طول یک مسیر در فضای حالت (ارتفاع درخت جستجو)

ارزیابی جستجوی سطحی

۱۶

□ س. جستجوی سطحی کدام گره‌ها را گسترش می‌دهد؟
□ تمام گره‌های بالاتر از کم عمق‌ترین گره راه‌حل!



1

b

b^2

b^d

ارزیابی جستجوی سطحی

۱۷

□ کامل؟ بله [اگر فاکتور انشعاب محدود باشد]

□ بهینه؟ خیر [مگر این که تابع هزینه مسیر برحسب عمق گره‌ها غیرنزولی باشد]

□ پیچیدگی زمانی؟ نمایی

$$b^1 + b^2 + b^3 + \dots + b^d \in O(b^d)$$

□ پیچیدگی حافظه؟ نمایی

$$O(b^{d-1}) + O(b^d) \in O(b^d)$$

↑
تعداد گره های
explored

↑
تعداد گره های
frontier

ارزیابی جستجوی سطحی

۱۸

□ مصرف زمان و حافظه در جستجوی سطحی.

عمق	تعداد گره ها	زمان	حافظه
۲	۱۱۰	۰ / ۱۱ میلی ثانیه	۱۰۷ کیلو بایت
۴	۱۱۱۱۰	۱۱ میلی ثانیه	۱۰/۶ مگابایت
۶	۱۰۶	۱/۱ ثانیه	۱ گیگابایت
۸	۱۰۸	۲ دقیقه	۱۰۳ گیگابایت
۱۰	۱۰۱۰	۳ ساعت	۱۰ ترابایت
۱۲	۱۰۱۲	۱۳ روز	۱ پتابایت
۱۴	۱۰۱۴	۳/۵ سال	۹۹ پتابایت
۱۶	۱۰۱۶	۳۵۰ سال	۱۰ هگزابایت

فاکتور انشعاب = ۱۰؛ یک میلیون گره بر ثانیه؛ ۱۰۰۰ بایت به ازای هر گره

جستجوی سطحی: مثال

۱۹

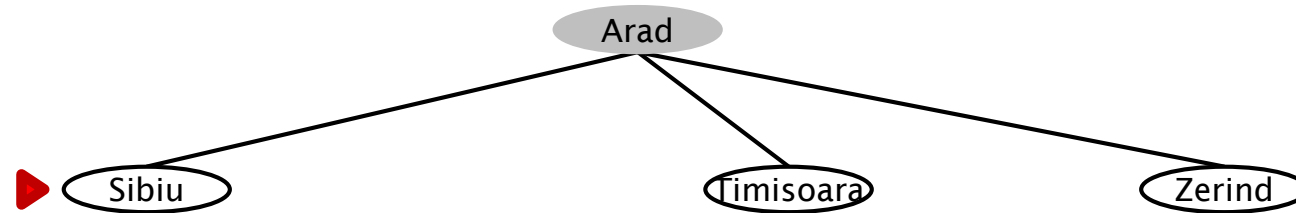


frontier: [Arad]

explored: {}

جستجوی سطحی: مثال

۲۰

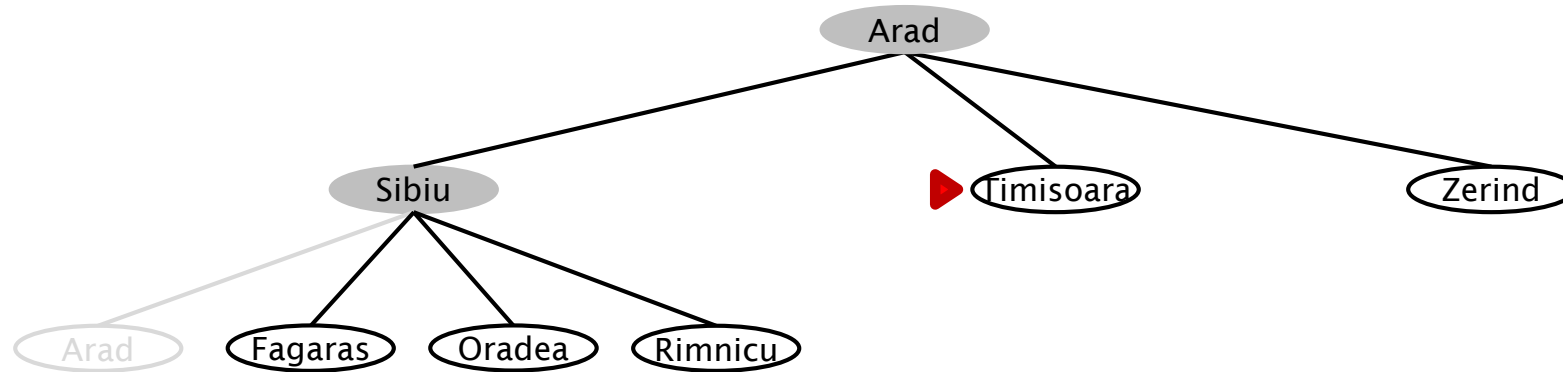


frontier: [Sibiu, Timisoara, Zerind]

explored: {Arad}

جستجوی سطحی: مثال

۲۱

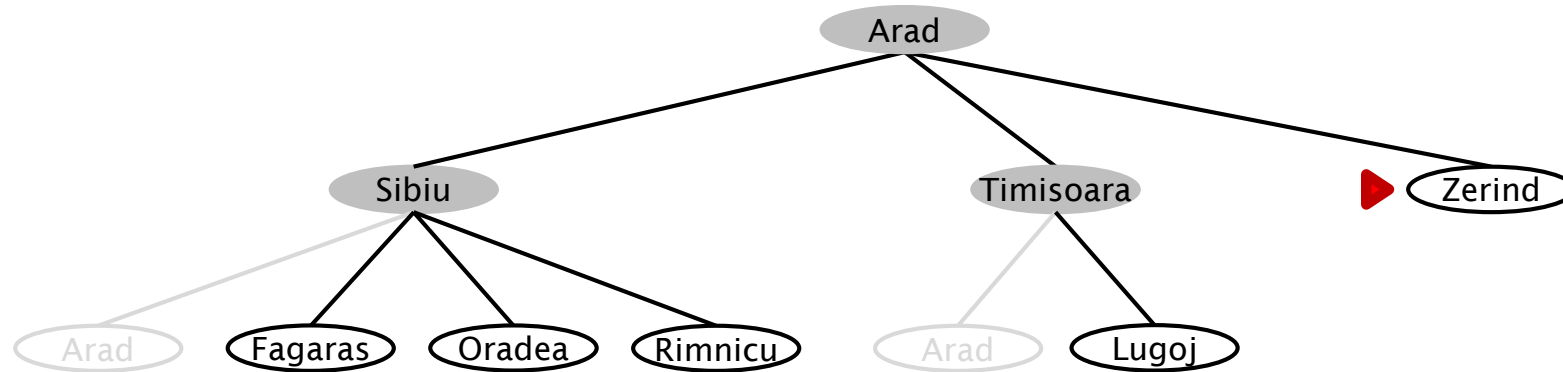


frontier: [Timisoara, Zerind, Fagaras, Oradea, Rimnicu]

explored: {Arad, Sibiu}

جستجوی سطحی: مثال

۲۲

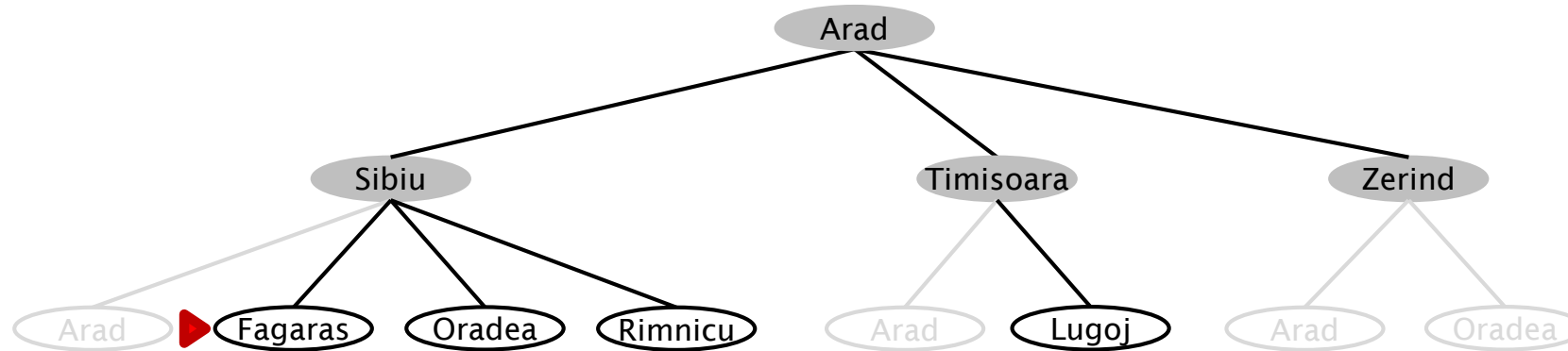


frontier: [Zerind, Fagaras, Oradea, Rimnicu, Lugoj]

explored: {Arad, Sibiu, Timisoara}

جستجوی سطحی: مثال

۲۳

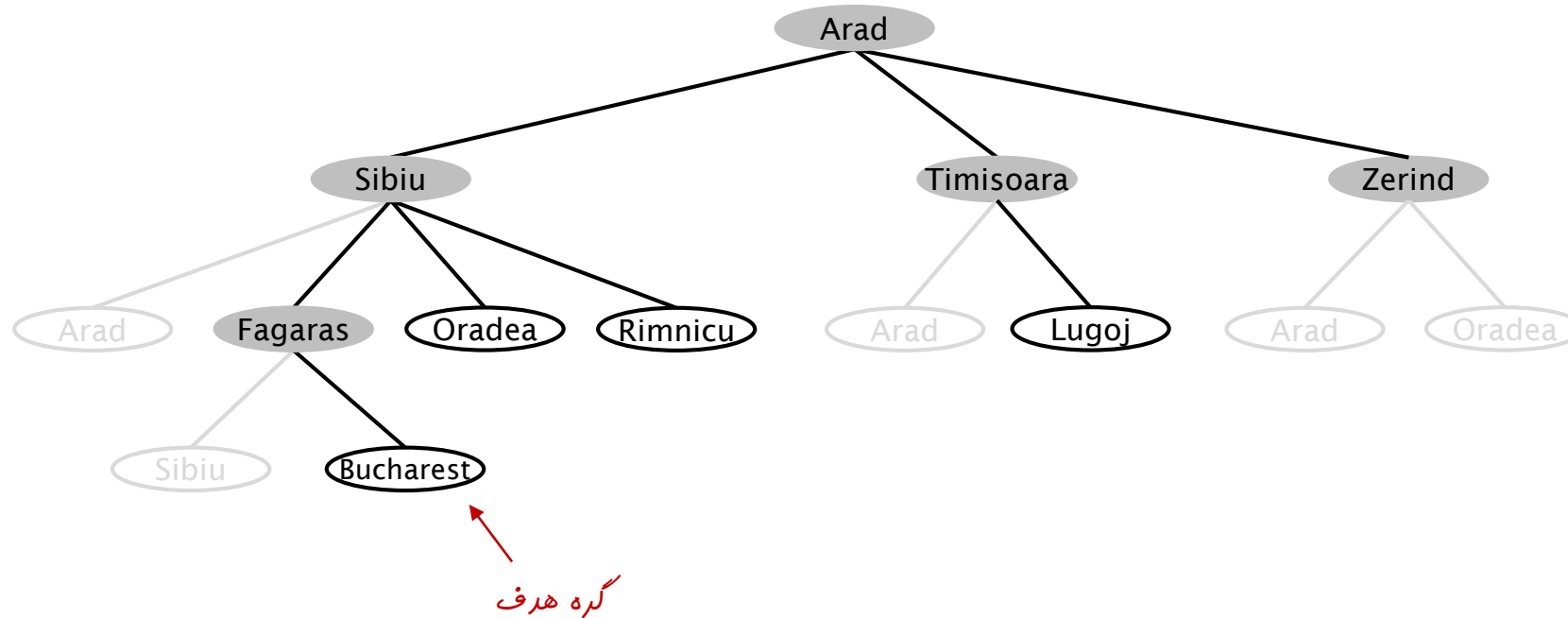


frontier: [Fagaras, Oradea, Rimnicu, Lugoj]

explored: {Arad, Sibiu, Timisoara, Zerind}

جستجوی سطحی: مثال

۲۴

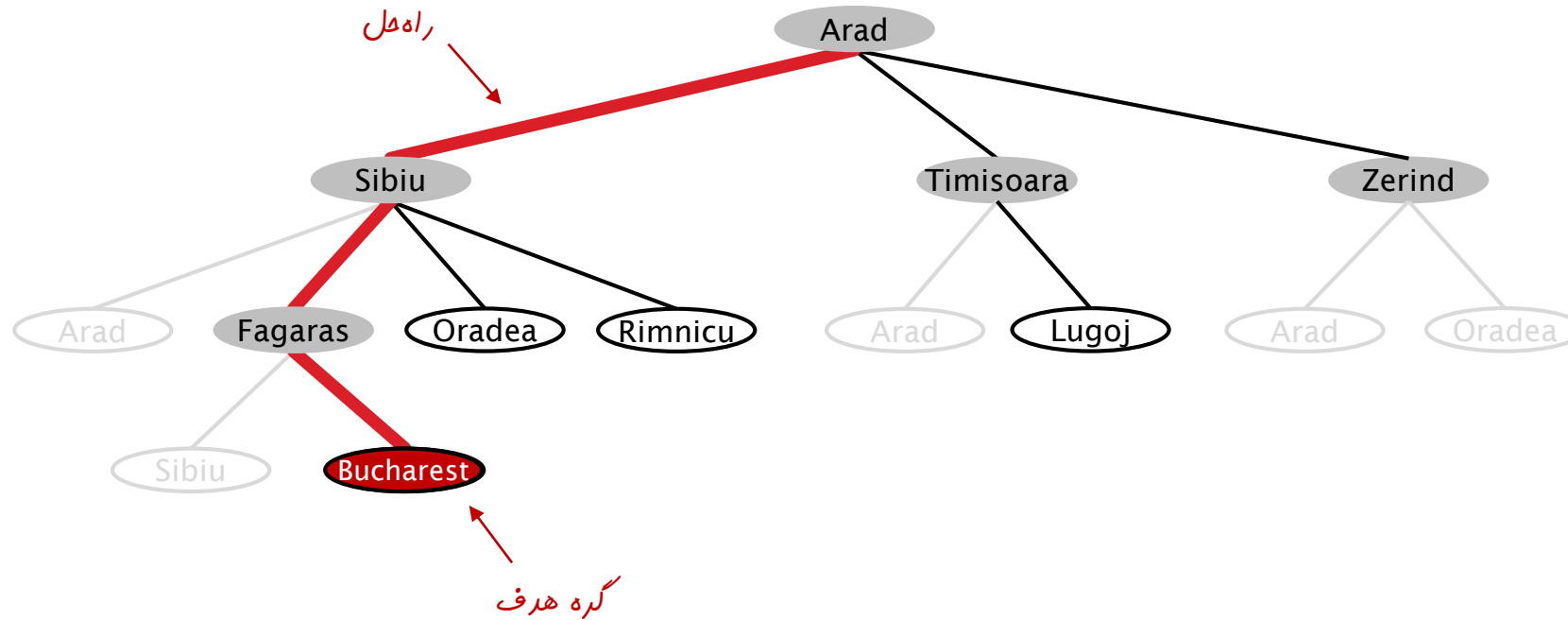


frontier: [Oradea, Rimnicu, Lugoj]

explored: {Arad, Sibiu, Timisoara, Zerind, Fagaras}

جستجوی سطحی: مثال

۲۵



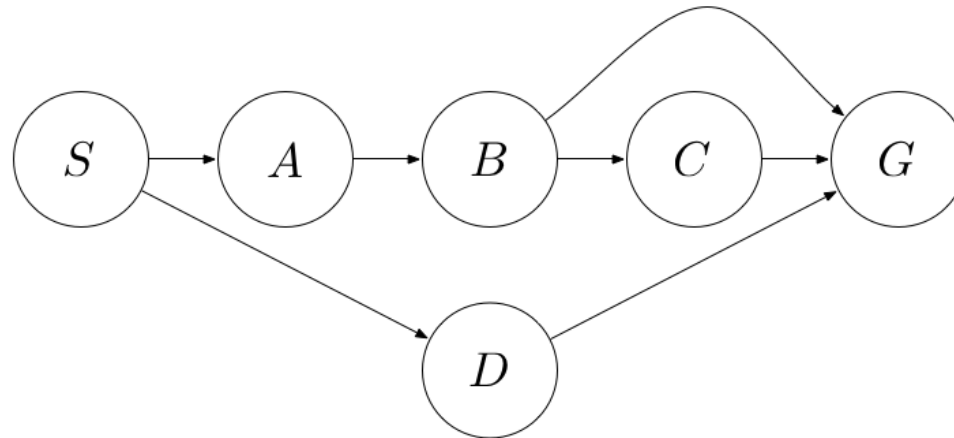
Sol. Arad → Sibiu → Fagaras → Bucharest

پرسش کلاسی: جستجوی سطحی درختی

۲۶

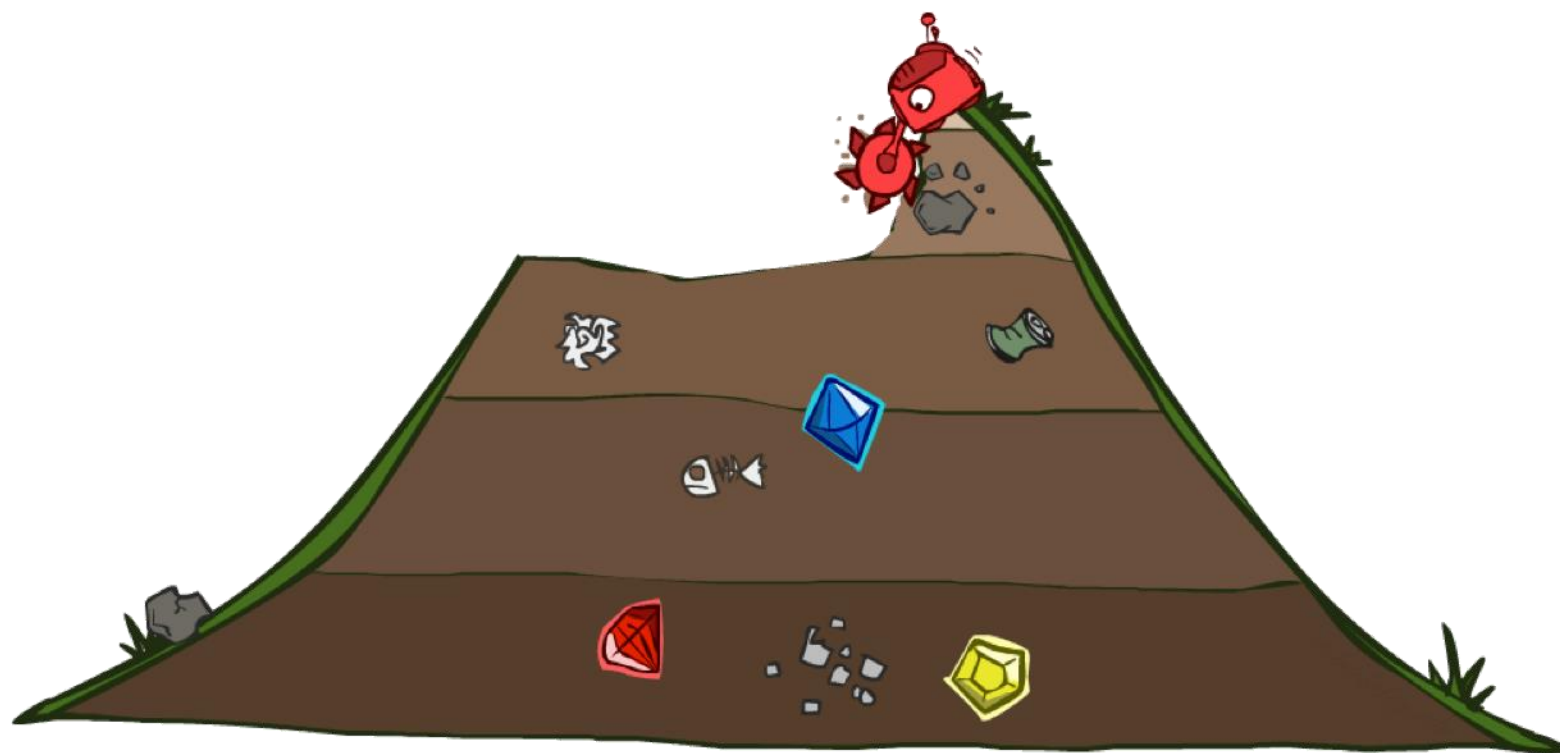
□ س. اگر بر روی گراف زیر یک جستجوی سطحی درختی انجام شود، چه مسیری به عنوان خروجی برگردانده می‌شود؟

□ فرض کنید در صورت وجود دو یا چند گره با عمق یکسان، گره‌ها به ترتیب الفبایی گسترش داده می‌شوند.



جستجوی هزینه یکنواخت

۲۷



جستجوی هزینه یکنواخت

۲۸

□ جستجوی هزینه یکنواخت.

□ هر بار **کم هزینه ترین** گره گسترش نیافته را گسترش می دهد.

□ هزینه گره: هزینه مسیر از ریشه تا آن گره در درخت جستجو.

□ پیاده سازی.

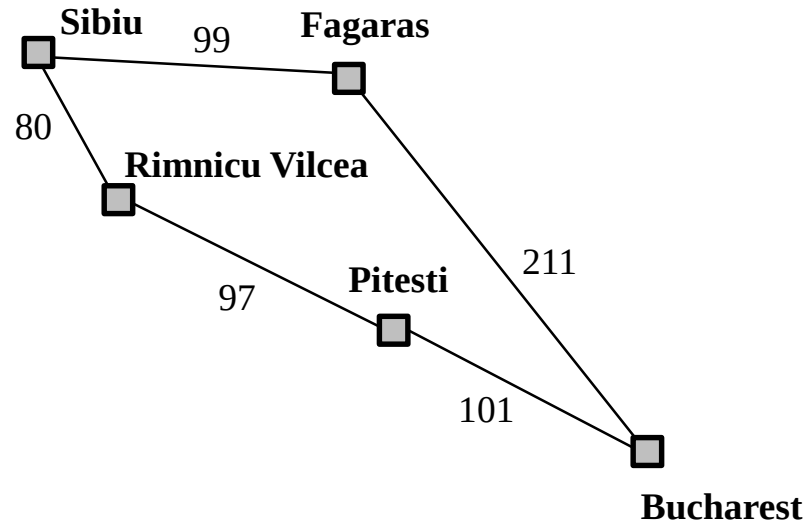
□ fringe یک صف اولویت است.

□ اولویت یک گره: هزینه مسیر از ریشه تا آن گره.

جستجوی هزینه یکنواخت

۲۹

□ مثال. یافتن مسیر از سیبوی به بخارست.



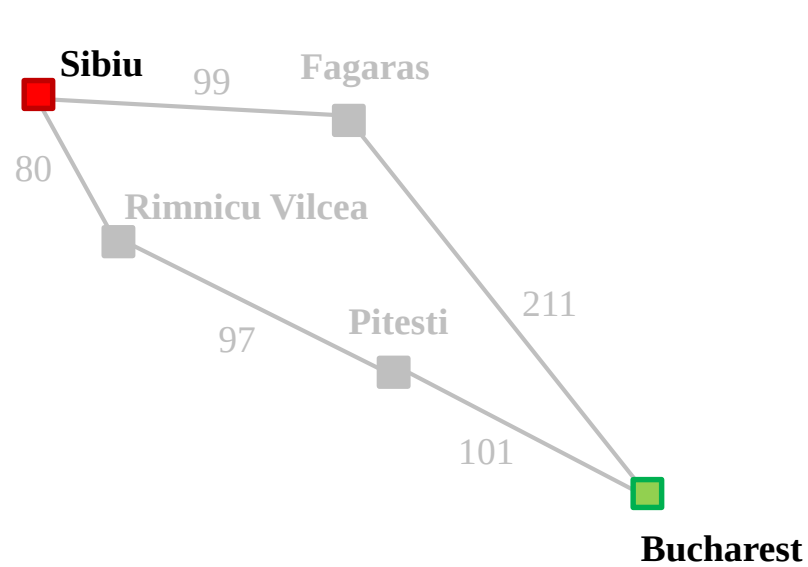
frontier: []

explored: {}

جستجوی هزینه یکنواخت

۳۰

□ مثال. یافتن مسیر از سیبوی به بخارست.



ایجاد ریشه درخت جستجو
با استفاده از حالت شروع

(S)
0

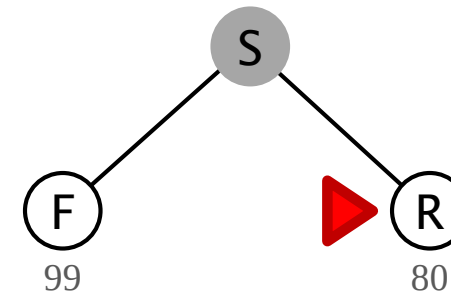
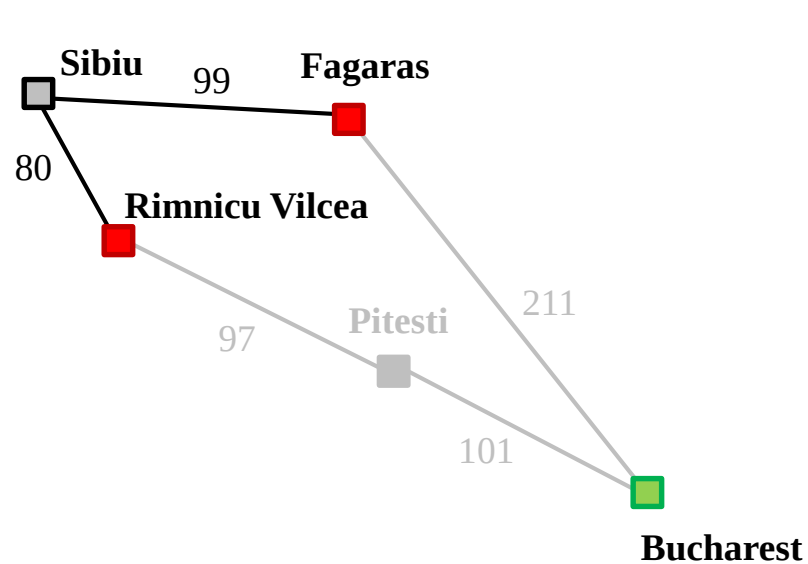
frontier: [Sibiu (0)]

explored: {}

جستجوی هزینه یکنواخت

۳۱

□ مثال. یافتن مسیر از سیبوی به بخارست.



گسترش گره ریشه و افزودن
فرزندان آن به صف اولویت

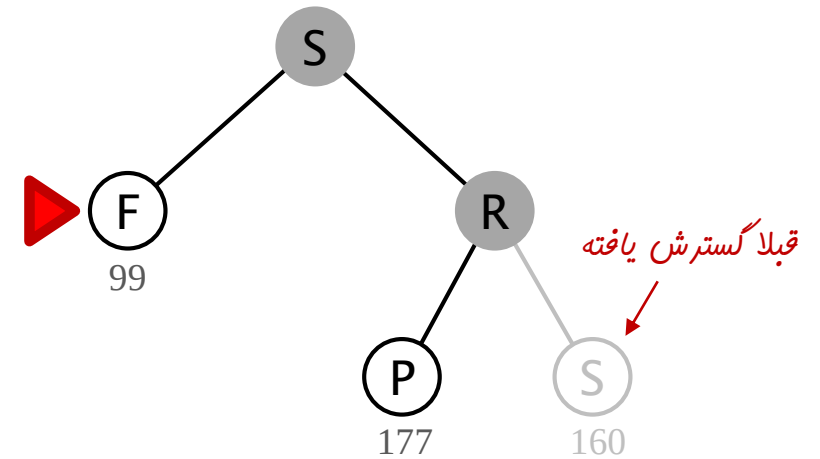
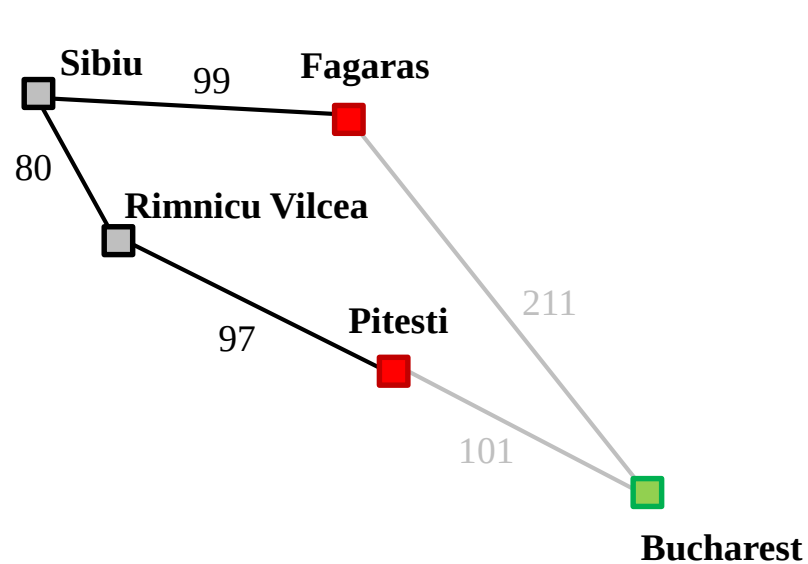
frontier: [Rimnicu (80), Fagaras (99)]

explored: {Sibiu}

جستجوی هزینه یکنواخت

۳۲

□ مثال. یافتن مسیر از سیبوی به بخارست.



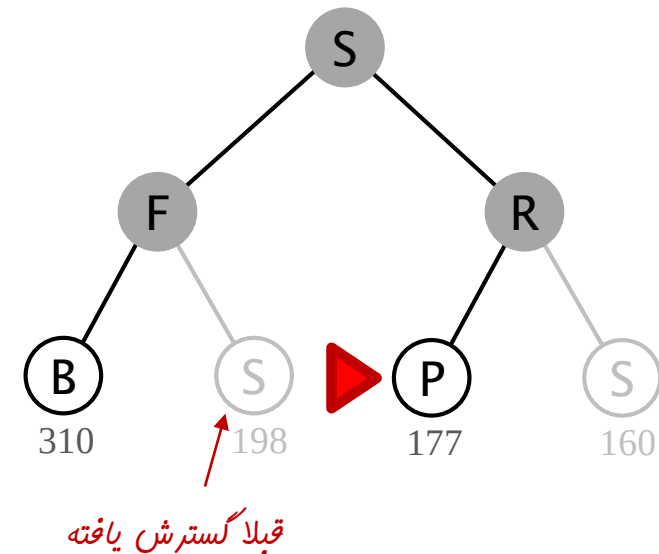
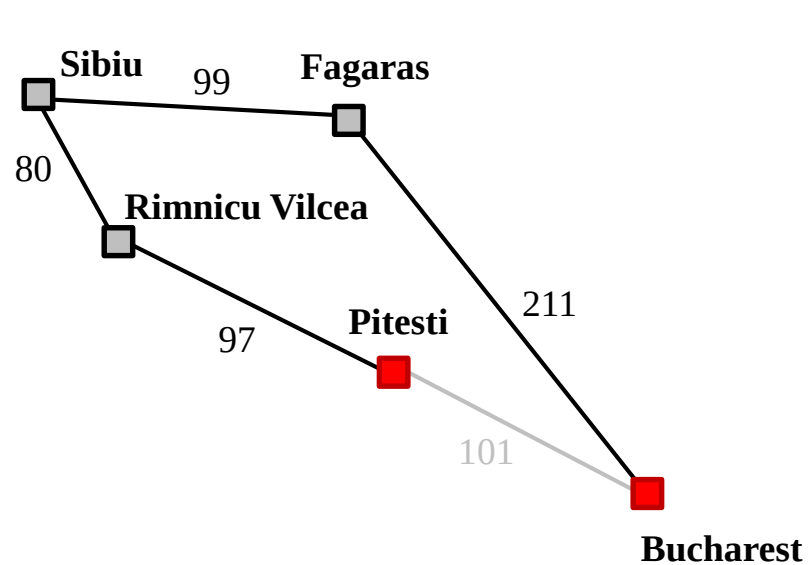
frontier: [Fagaras (99), Pitesti (177)]

explored: {Sibiu, Rimnicu}

جستجوی هزینه یکنواخت

۳۳

□ مثال. یافتن مسیر از سیبوی به بخارست.



قبلا گسترش یافته

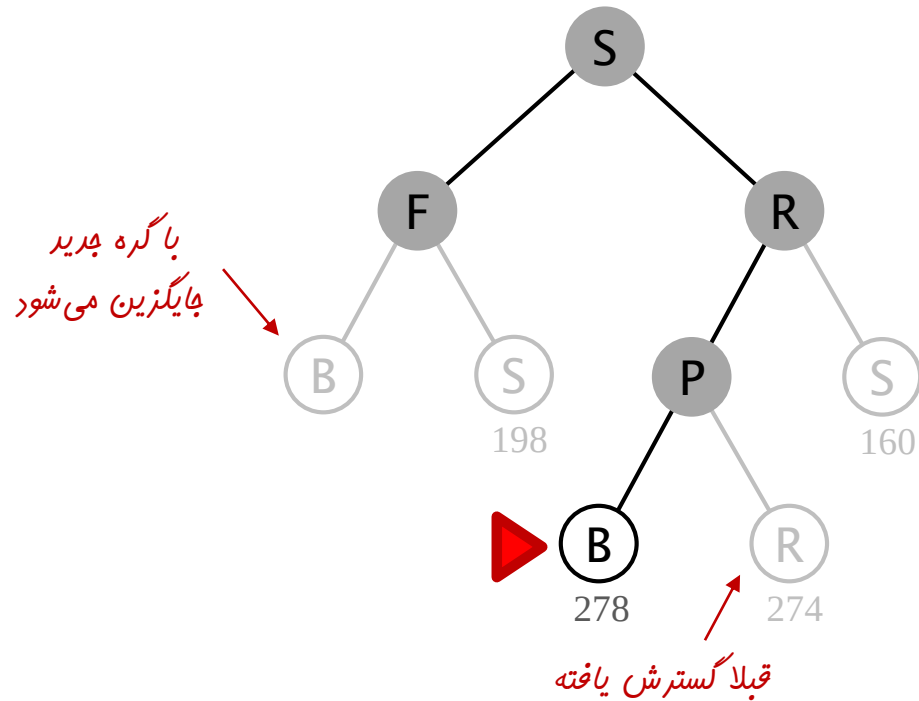
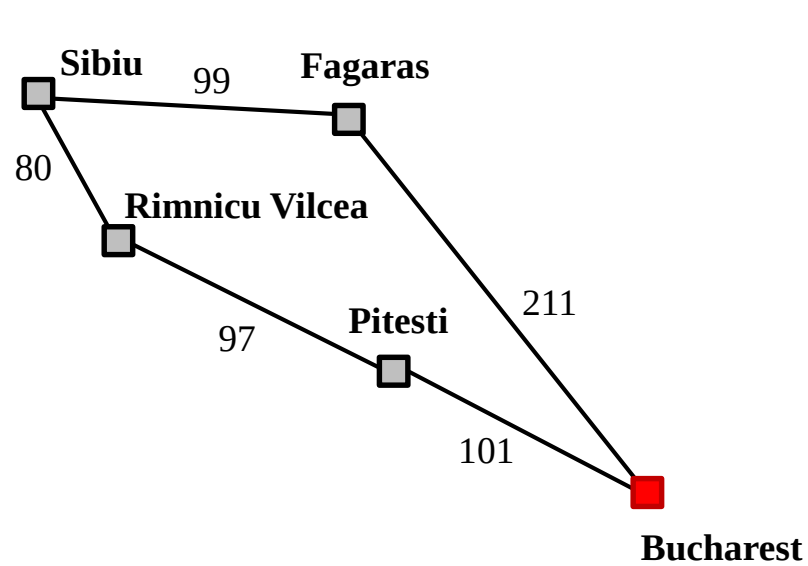
frontier: [Pitesti (177), Bucharest (310)]

explored: {Sibiu, Rimnicu, Fagaras}

جستجوی هزینه یکنواخت

۳۴

□ مثال. یافتن مسیر از سیبوی به بخارست.



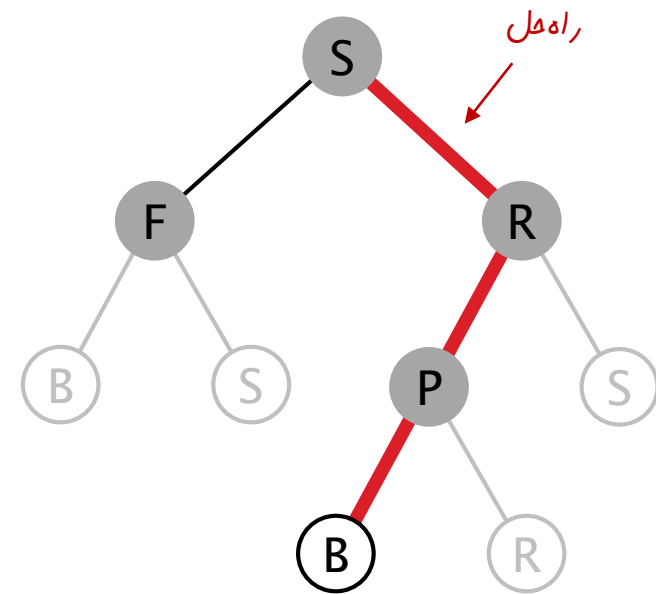
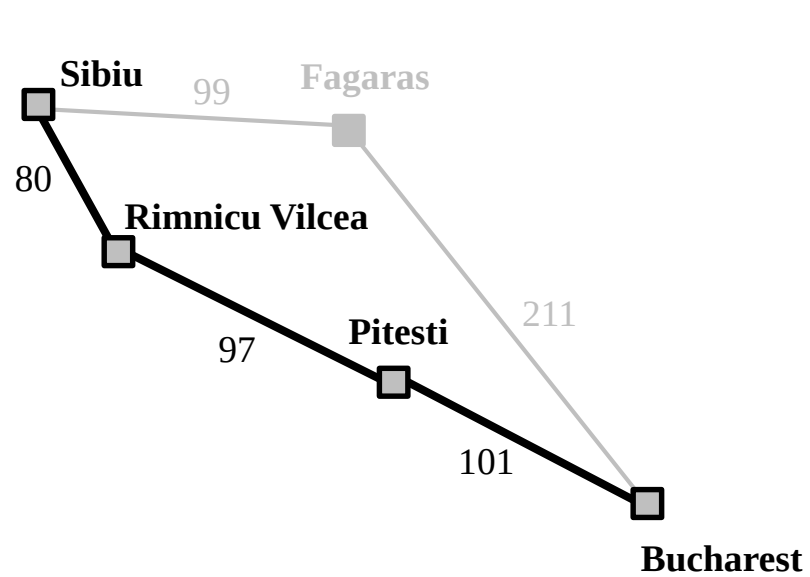
frontier: [Bucharest (278)]

explored: {Sibiu, Rimnicu, Fagaras, Pitesti}

جستجوی هزینه یکنواخت

۳۵

□ مثال. یافتن مسیر از سیبوی به بخارست.



جستجوی هزینه یکنواخت

۳۶

```
function UNIFORM-COST-SEARCH(problem) returns a solution, or failure
  node ← a node with STATE = problem.INITIAL-STATE, PATH-COST = 0
  frontier ← a priority queue ordered by PATH-COST, with node as the only element
  explored ← an empty set
  loop do
    if EMPTY?(frontier) then return failure
    node ← POP(frontier) /* chooses the lowest-cost node in frontier */
    if problem.GOAL-TEST(node.STATE) then return SOLUTION(node)
    add node.STATE to explored
    for each action in problem.ACTIONS(node.STATE) do
      child ← CHILD-NODE(problem, node, action)
      if child.STATE not in explored or frontier then
        frontier ← INSERT(child, frontier)
      else if child.STATE is in frontier with higher PATH-COST then
        replace that frontier node with child
```

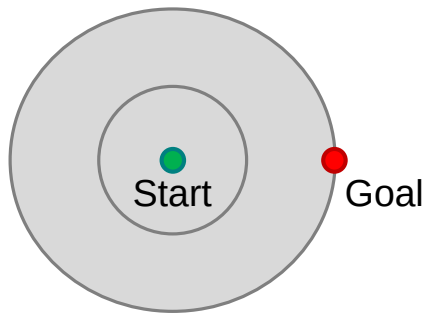
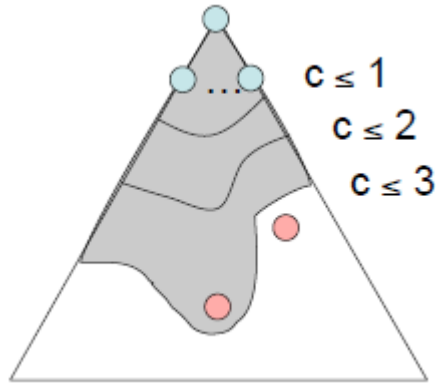
انتخاب

آزمایش هدف

گسترش

ارزیابی جستجوی هزینه یکنواخت

۳۷



جستجوی هزینه یکسان بدون توجه به هدف،
در همه جهات ها جستجو می کند.

□ کامل؟ بله [اگر هزینه تمام عمل ها مثبت باشد]

□ بهینه؟ بله [اگر هزینه تمام عمل ها مثبت باشد]

$$O(b^{1+\lceil C^*/\epsilon \rceil})$$

□ پیچیدگی زمانی؟ نمایی

$$O(b^{1+\lceil C^*/\epsilon \rceil})$$

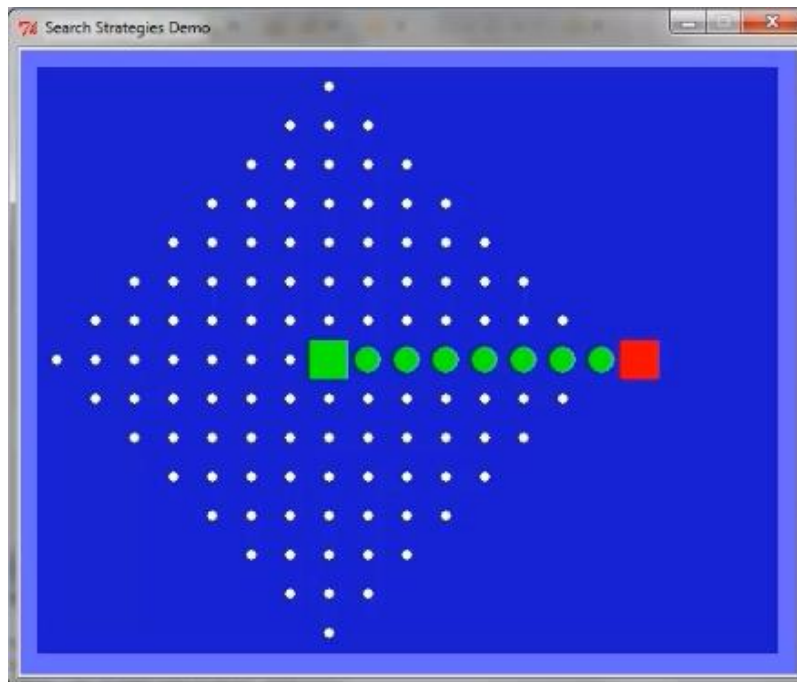
□ پیچیدگی حافظه؟ نمایی

ارزیابی جستجوی هزینه یکنواخت

۳۸

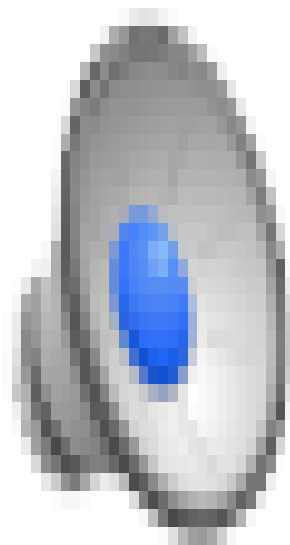
□ جستجوی هزینه یکنواخت.

□ بررسی حالت‌ها در تمام جهت‌ها برای پیدا کردن راه‌حل بهینه.



جستجوی هزینه یکنواخت: اجرای نمایشی

۳۹

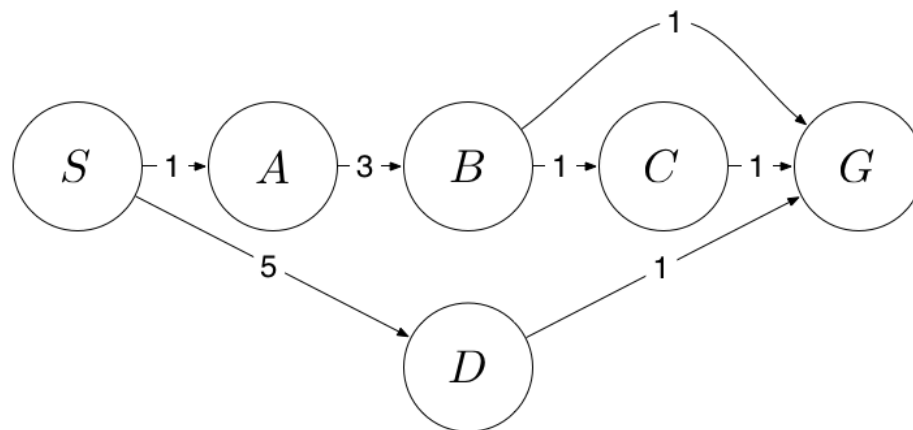


پرسش کلاسی: جستجوی هزینه یکنواخت

۴۰

□ س. اگر بر روی گراف زیر یک جستجوی هزینه یکنواخت درختی انجام شود، چه مسیری به عنوان خروجی برگردانده می‌شود؟

□ فرض کنید در صورت وجود دو یا چند گره با هزینه یکسان، گره‌ها به ترتیب الفبایی گسترش داده می‌شوند.



استراتژی‌های جستجوی سطحی و هزینه یکنواخت

۴۱

□ س. بزرگ‌ترین مشکل استراتژی‌های جستجوی سطحی و هزینه یکنواخت چیست؟



□ ایده. در هر لحظه از جستجو تنها یک مسیر را کاوش و در حافظه نگهداری کن.

□ استراتژی جستجوی عمقی!

جستجوی عمقی

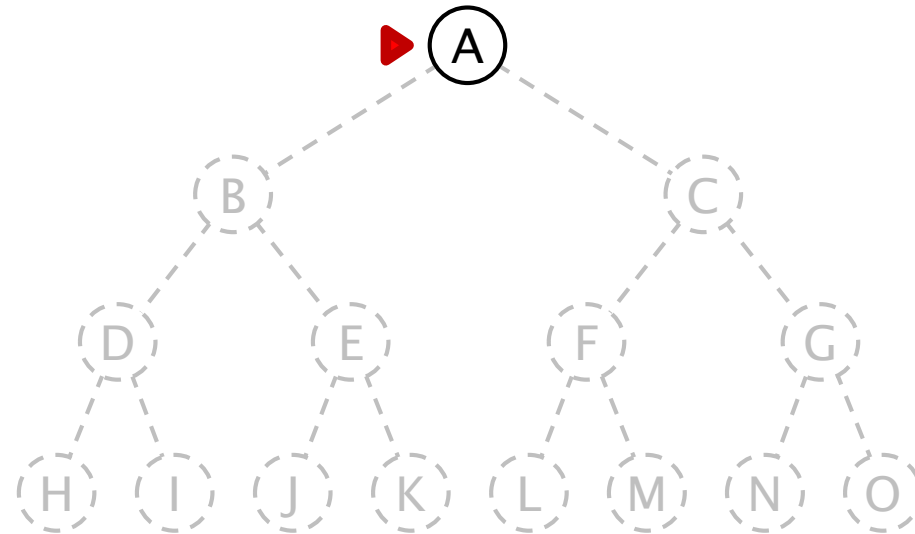
۴۲



جستجوی عمقی

۴۳

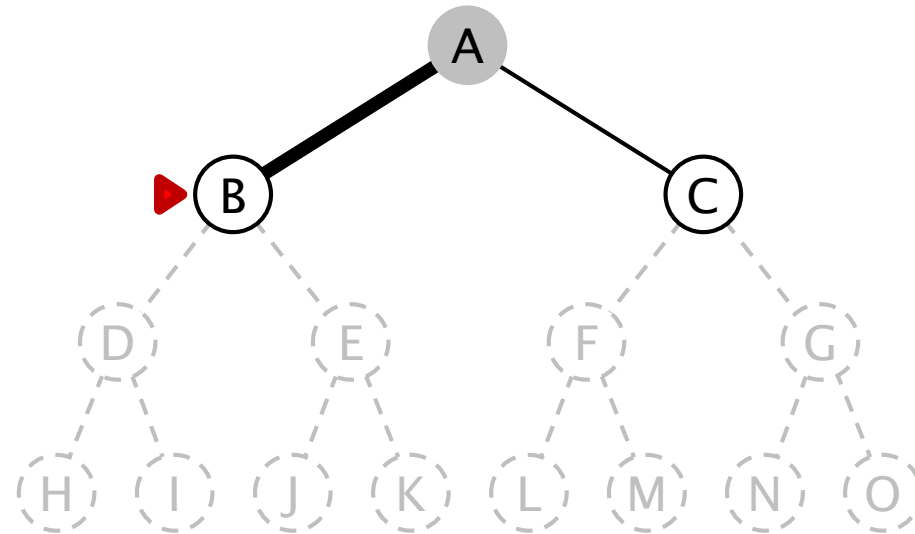
- جستجوی عمقی. هر بار **عمیق‌ترین** گره گسترش نیافته را گسترش می‌دهد.
- پیاده‌سازی. با استفاده از پشته؛ درج فرزندان جدید در ابتدای لیست.



جستجوی عمقی

۴۴

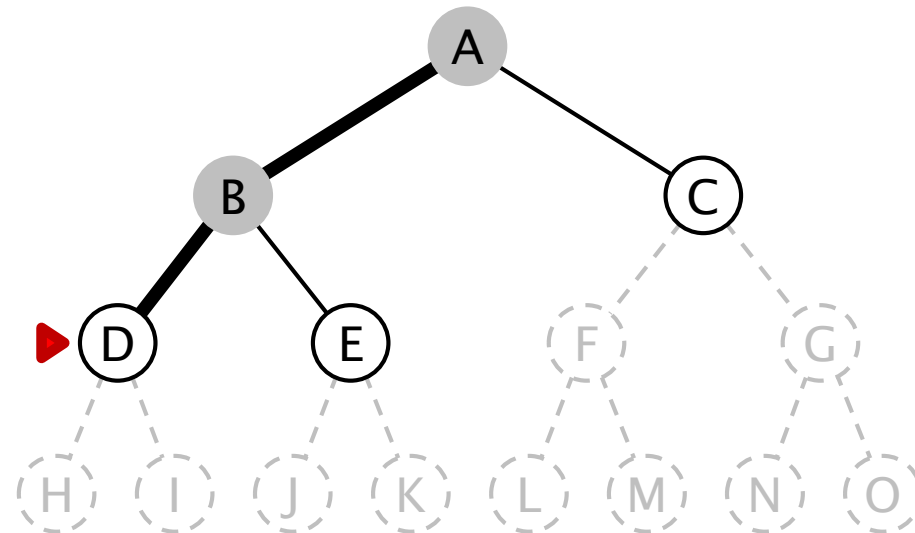
- جستجوی عمقی. هر بار **عمیق‌ترین** گره گسترش نیافته را گسترش می‌دهد.
- پیاده‌سازی. با استفاده از پشته؛ درج فرزندان جدید در ابتدای لیست.



جستجوی عمقی

۴۵

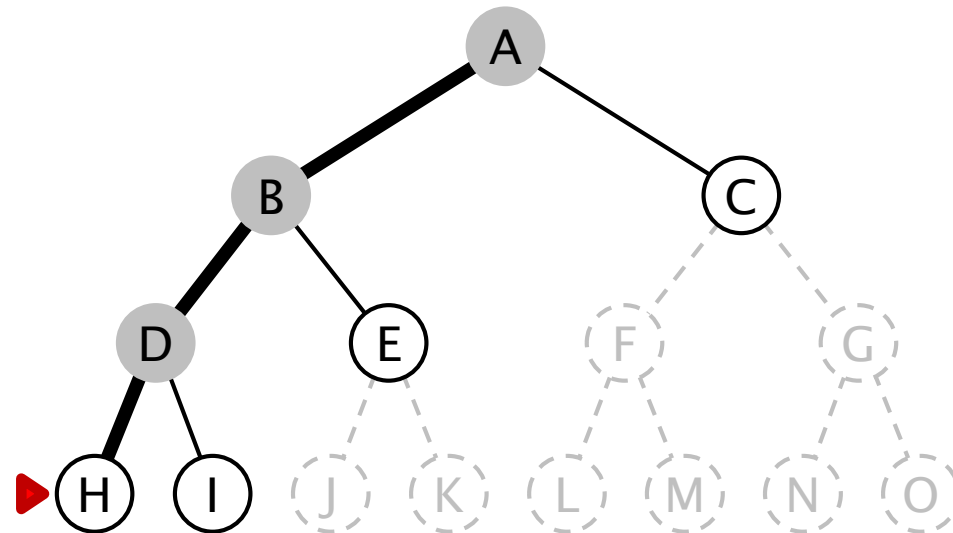
- جستجوی عمقی. هر بار **عمیق‌ترین** گره گسترش نیافته را گسترش می‌دهد.
- پیاده‌سازی. با استفاده از پشته؛ درج فرزندان جدید در ابتدای لیست.



جستجوی عمقی

۴۶

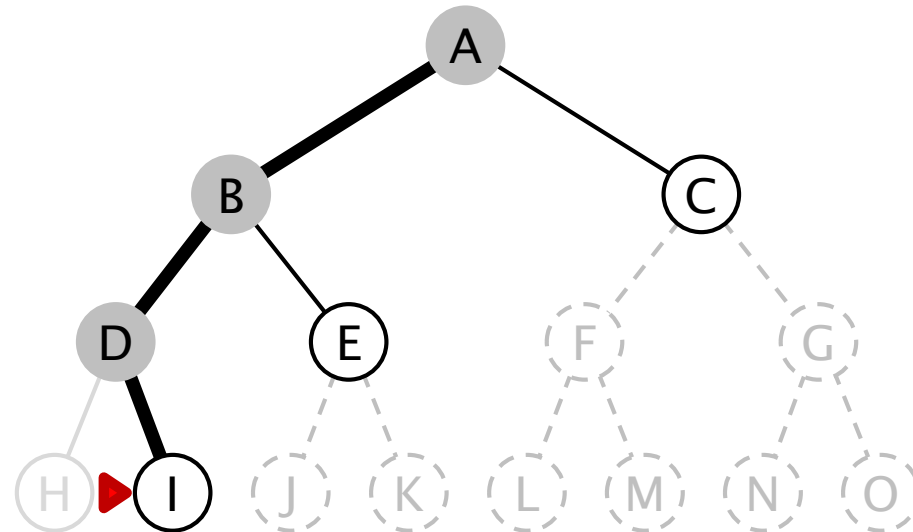
- جستجوی عمقی. هر بار **عمیق‌ترین** گره گسترش نیافته را گسترش می‌دهد.
- پیاده‌سازی. با استفاده از پشته؛ درج فرزندان جدید در ابتدای لیست.



جستجوی عمقی

۴۷

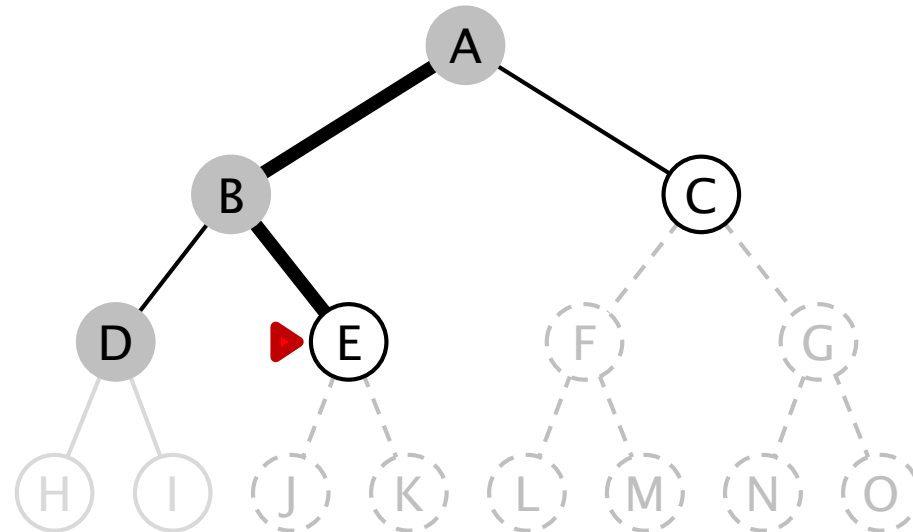
- جستجوی عمقی. هر بار **عمیق‌ترین** گره گسترش نیافته را گسترش می‌دهد.
- پیاده‌سازی. با استفاده از پشته؛ درج فرزندان جدید در ابتدای لیست.



جستجوی عمقی

۴۸

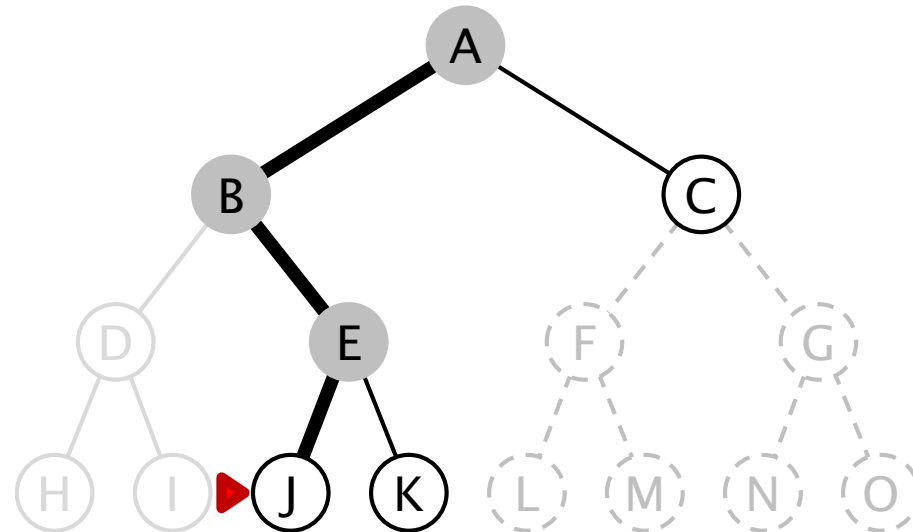
- جستجوی عمقی. هر بار **عمیق‌ترین** گره گسترش نیافته را گسترش می‌دهد.
- پیاده‌سازی. با استفاده از پشته؛ درج فرزندان جدید در ابتدای لیست.



جستجوی عمقی

۴۹

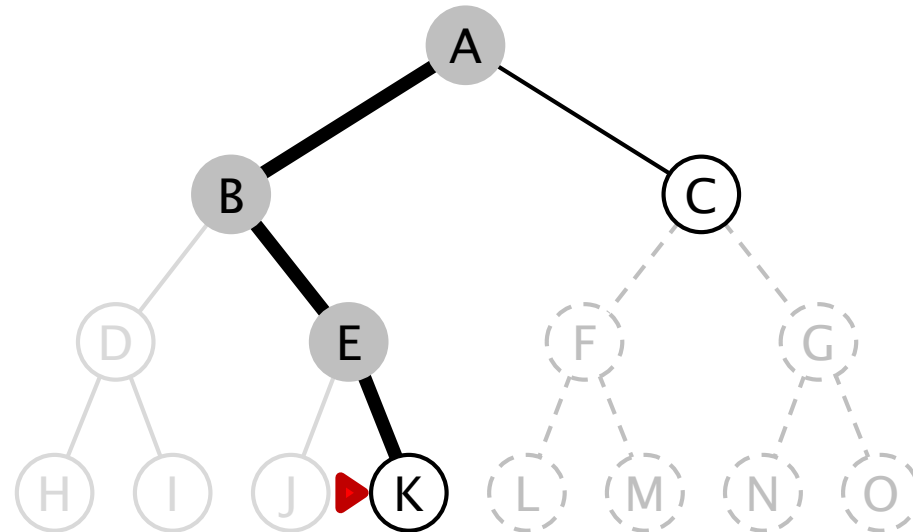
- جستجوی عمقی. هر بار **عمیق‌ترین** گره گسترش نیافته را گسترش می‌دهد.
- پیاده‌سازی. با استفاده از پشته؛ درج فرزندان جدید در ابتدای لیست.



جستجوی عمقی

۵۰

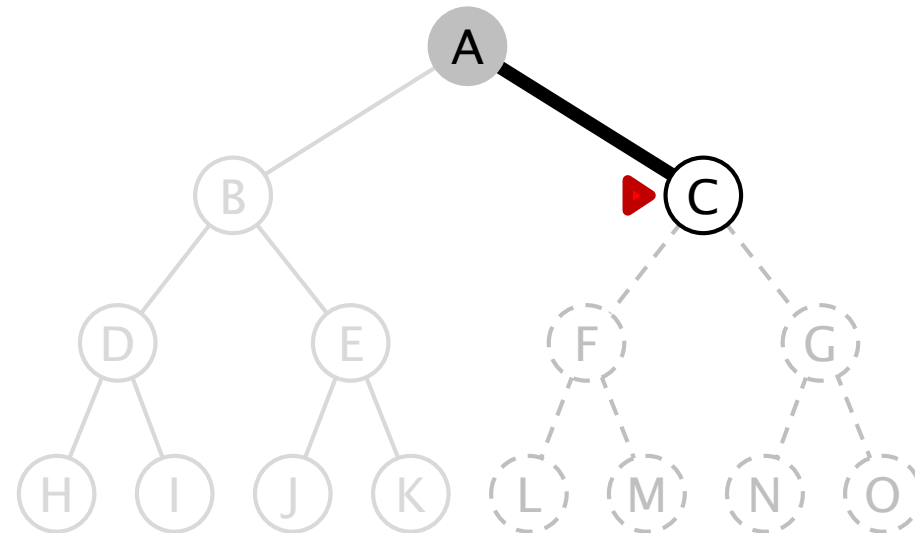
- جستجوی عمقی. هر بار **عمیق‌ترین** گره گسترش نیافته را گسترش می‌دهد.
- پیاده‌سازی. با استفاده از پشته؛ درج فرزندان جدید در ابتدای لیست.



جستجوی عمقی

۵۱

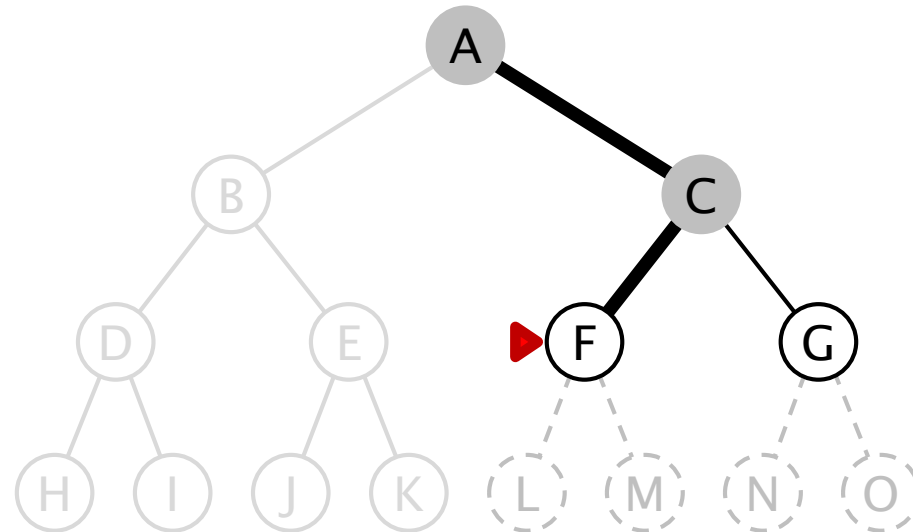
- جستجوی عمقی. هر بار **عمیق‌ترین** گره گسترش نیافته را گسترش می‌دهد.
- پیاده‌سازی. با استفاده از پشته؛ درج فرزندان جدید در ابتدای لیست.



جستجوی عمقی

۵۲

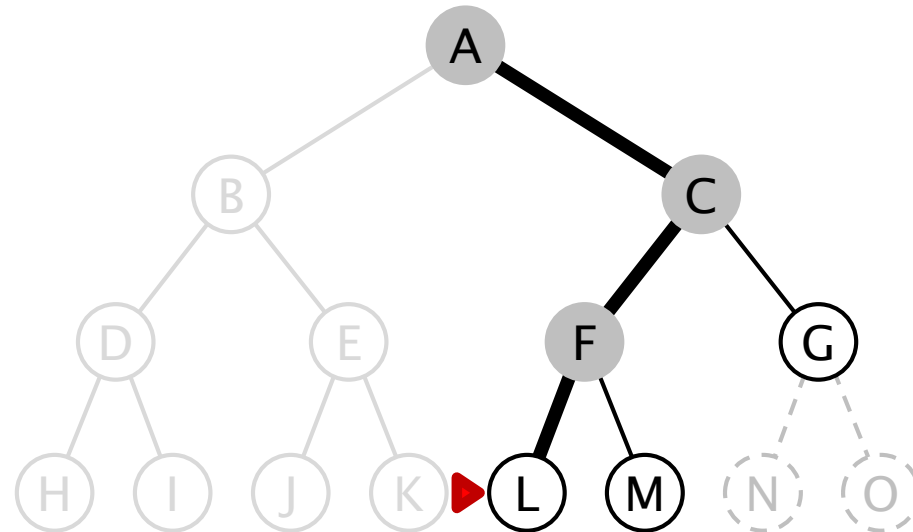
- جستجوی عمقی. هر بار **عمیق‌ترین** گره گسترش نیافته را گسترش می‌دهد.
- پیاده‌سازی. با استفاده از پشته؛ درج فرزندان جدید در ابتدای لیست.



جستجوی عمقی

۵۳

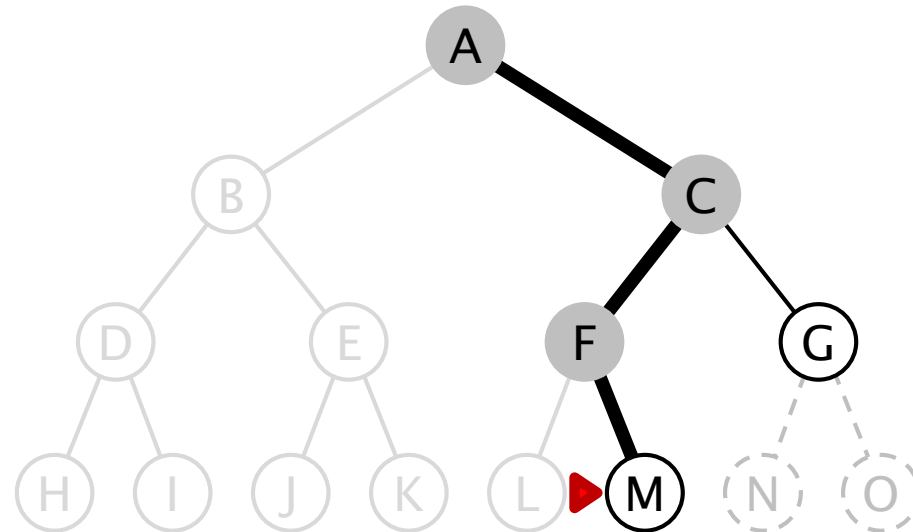
- جستجوی عمقی. هر بار **عمیق‌ترین** گره گسترش نیافته را گسترش می‌دهد.
- پیاده‌سازی. با استفاده از پشته؛ درج فرزندان جدید در ابتدای لیست.



جستجوی عمقی

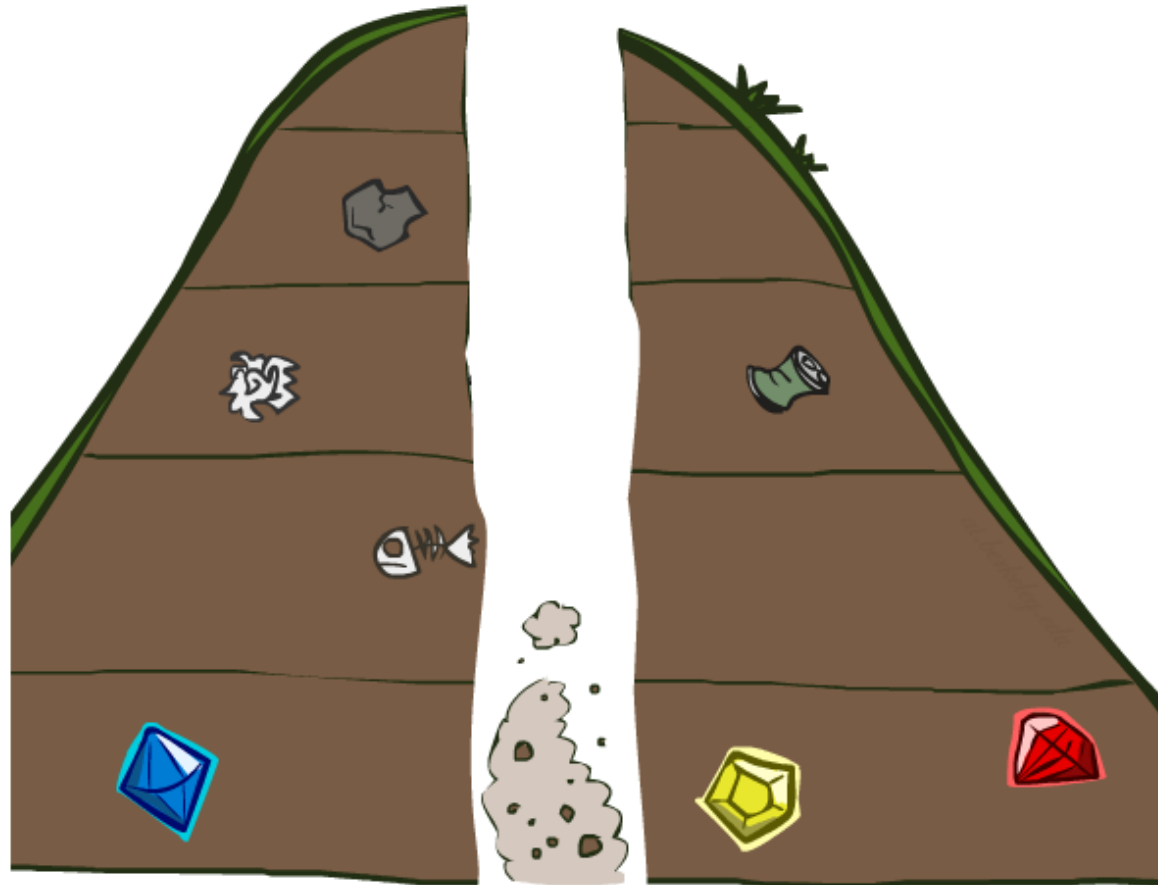
۵۴

- جستجوی عمقی. هر بار **عمیق‌ترین** گره گسترش نیافته را گسترش می‌دهد.
- پیاده‌سازی. با استفاده از پشته؛ درج فرزندان جدید در ابتدای لیست.



ارزیابی جستجوی عمقی

۵۵

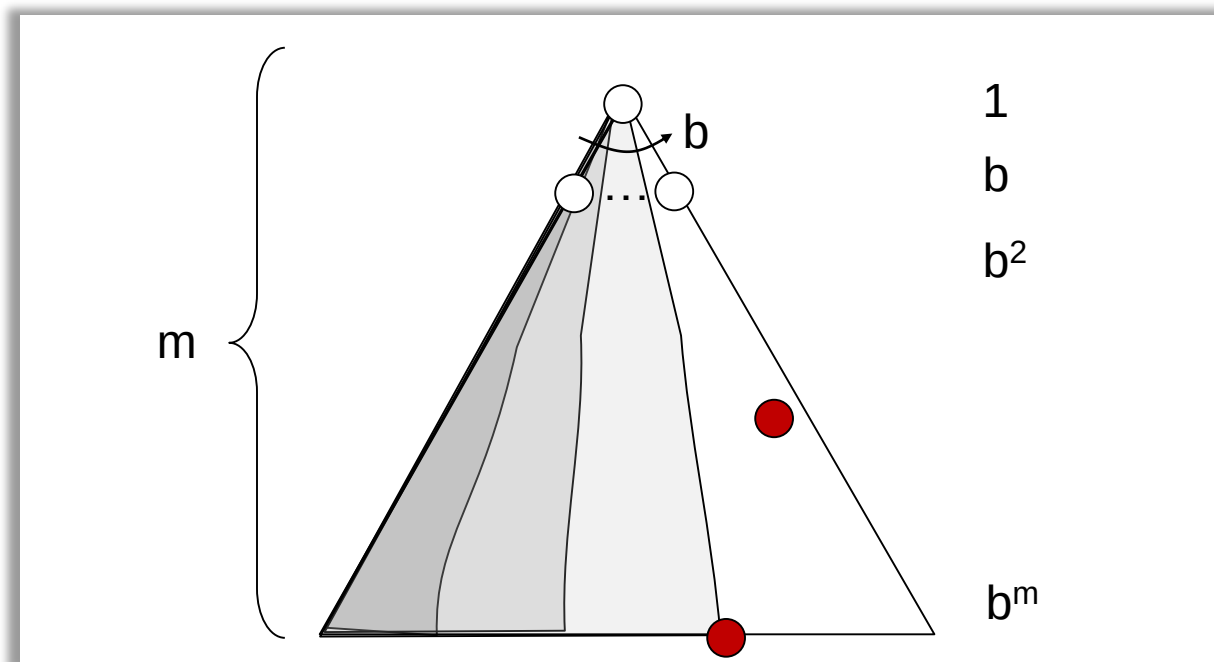


ارزیابی جستجوی عمقی

۵۶

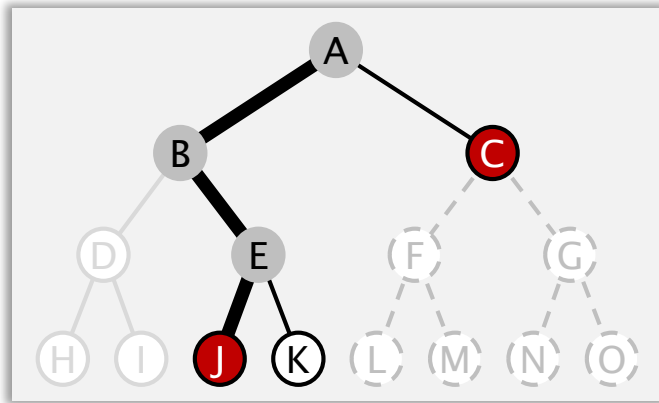
□ جستجوی عمقی کدام گره‌ها را گسترش می‌دهد؟

□ تمام گره‌های قبل از چپ‌ترین گره راه‌حل.



ارزیابی جستجوی عمقی

۵۷



□ کامل؟

□ جستجوی درختی: خیر

□ جستجوی گراف‌ی: بله [فضای حالت متناهی]

□ بهینه؟ خیر [شکل روبه‌رو]

□ پیچیدگی زمانی؟ نمایی

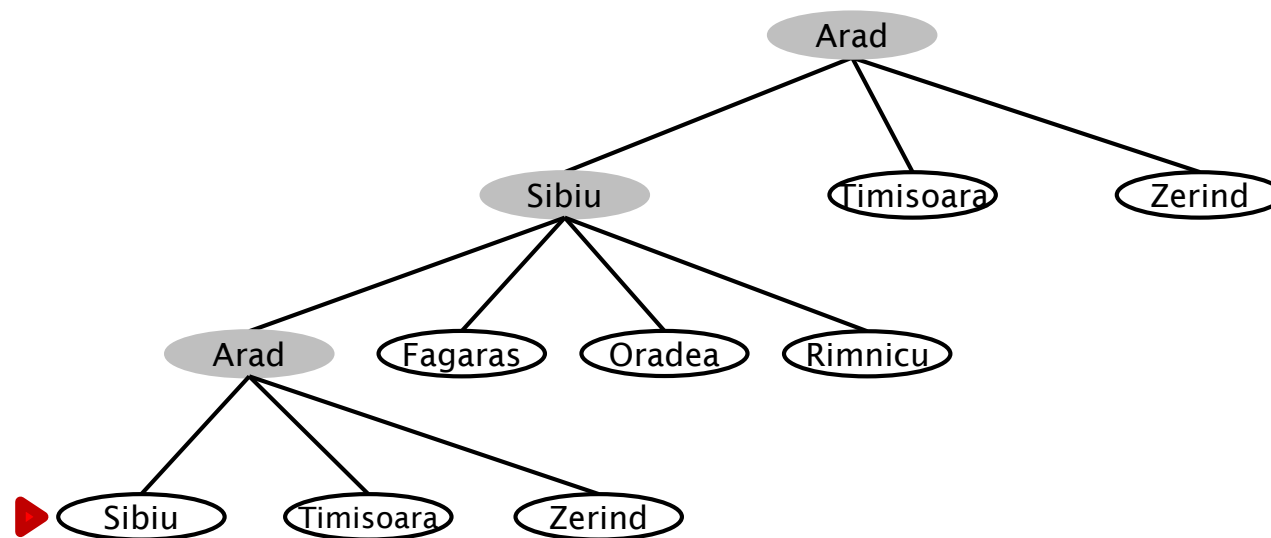
□ پیچیدگی حافظه؟ خطی [جستجوی درختی]

$$b^1 + b^2 + b^3 + \dots + b^m \in O(b^m)$$

$$O(bm)$$

جستجوی عمقی: افتادن در حلقه بینهایت

۵۸

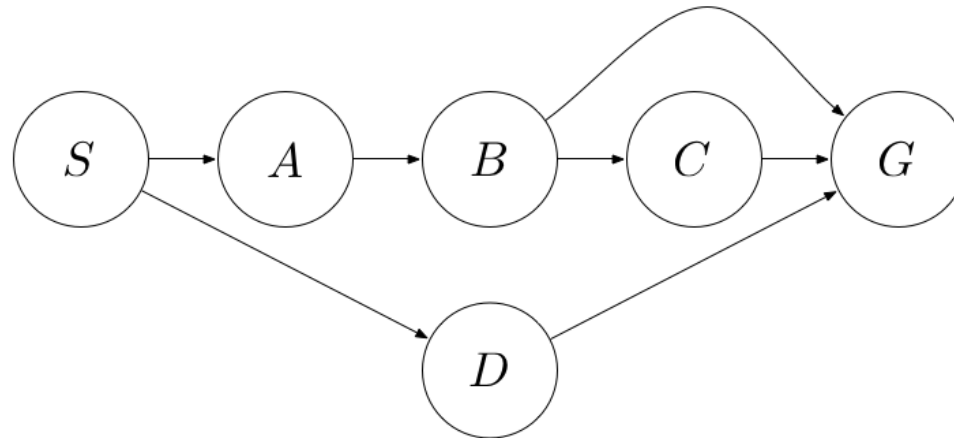


پرسش کلاسی: جستجوی عمقی درختی

۵۹

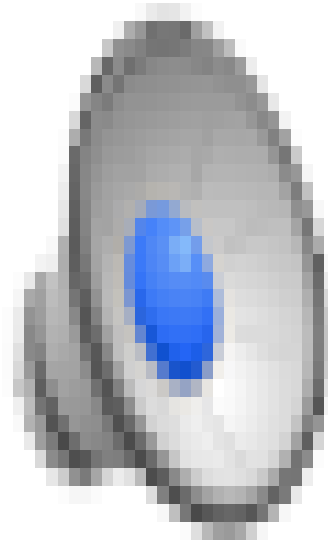
□ س. اگر بر روی گراف زیر یک جستجوی عمقی انجام شود، چه مسیری به عنوان خروجی برگردانده می‌شود؟

□ فرض کنید در صورت وجود دو یا چند گره با عمق یکسان، گره‌ها به ترتیب الفبایی گسترش داده می‌شوند.



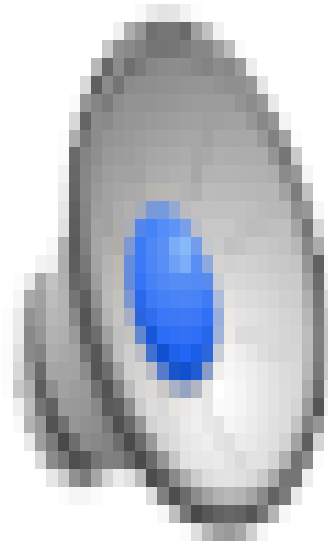
پرسش کلاسی: کدام استراتژی؟

۶۰



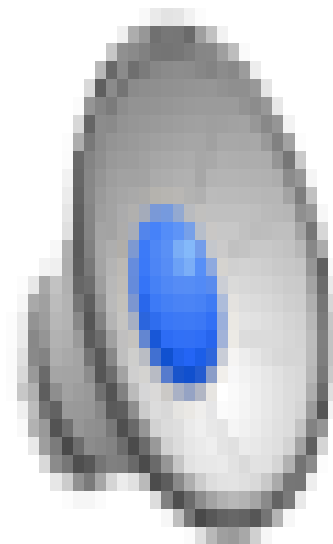
پرسش کلاسی: کدام استراتژی؟

۶۱



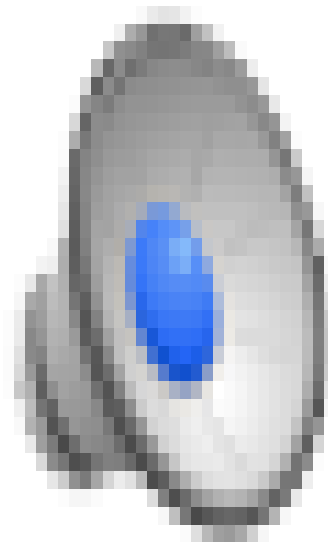
پرسش کلاسی: کدام استراتژی؟

۶۲



پرسش کلاسی: کدام استراتژی؟

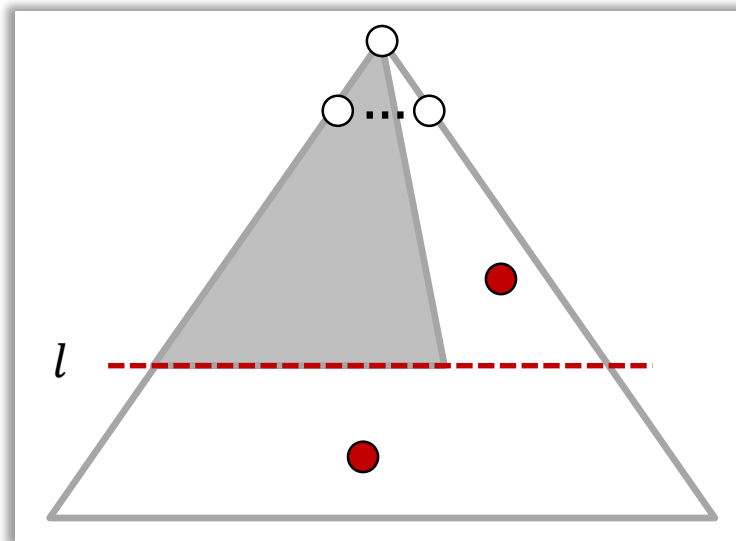
۶۳



جستجوی عمقی محدود

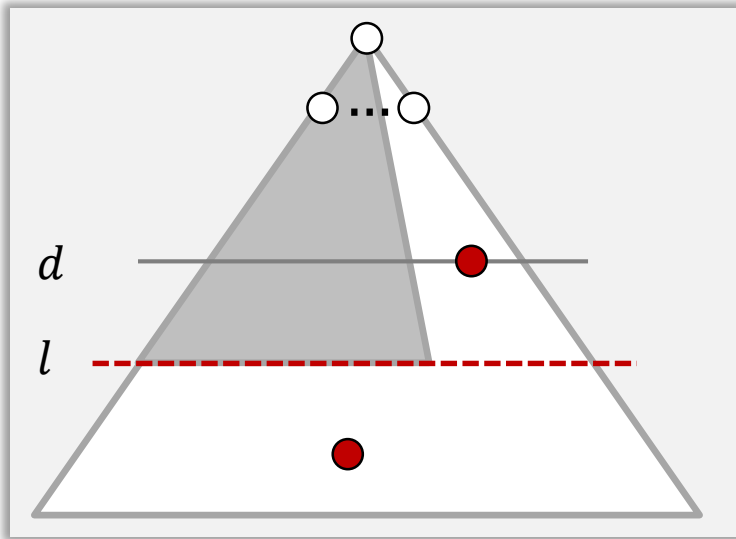
۶۴

- ایده. محدود کردن عمق جستجو در جستجوی عمقی.
- فرزندان گره‌های واقع در عمق l تولید نمی‌شوند.



ارزیابی جستجوی عمقی محدود

۶۵



□ کامل؟ بله [اگر $l \geq d$]

□ بهینه؟ خیر (همانند جستجوی عمقی)

□ پیچیدگی زمانی؟ نمایی

□ پیچیدگی حافظه؟ خطی [جستجوی درختی]

$$b^1 + b^2 + b^3 + \dots + b^l \in O(b^l)$$

$$O(bl)$$

```
function DEPTH-LIMITED-SEARCH(problem, limit) returns a solution, or failure/cutoff  
  return RECURSIVE-DLS(MAKE-NODE(problem.INITIAL-STATE), problem, limit)  
  
function RECURSIVE-DLS(node, problem, limit) returns a solution, or failure/cutoff  
  if problem.GOAL-TEST(node.STATE) then return SOLUTION(node)  
  else if limit = 0 return cutoff  
  else  
    cutoff_occured?  $\leftarrow$  false  
    for each action in problem.ACTIONS(node.STATE) do  
      child  $\leftarrow$  CHILD-NODE(problem, node, action)  
      result  $\leftarrow$  RECURSIVE-DLS(child, problem, limit - 1)  
      if result = cutoff then cutoff_occured?  $\leftarrow$  true  
      else if result  $\neq$  failure then return result  
  if cutoff_occured? then return cutoff else return failure
```

ایراد اصلی. تعیین یک محدوده عمقی مناسب!

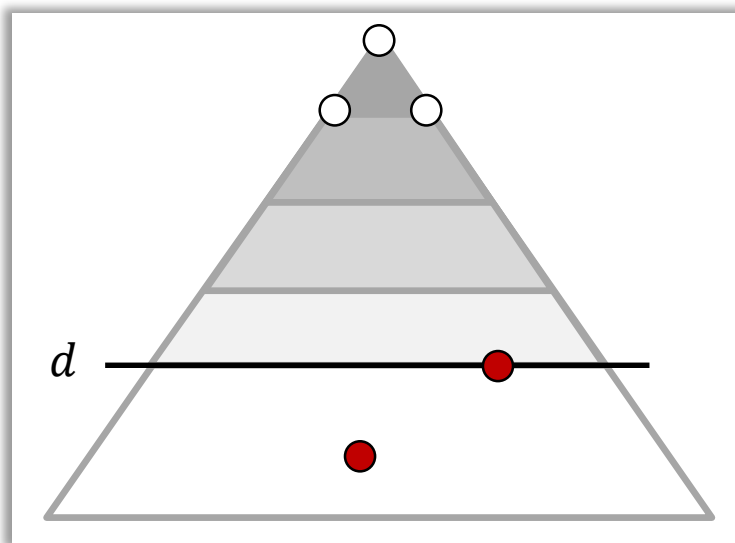
جستجوی عمیق کننده تکراری

۶۷

□ ایراد اصلی DLS. انتخاب یک محدوده عمقی مناسب.

□ جستجوی عمیق کننده تکراری.

□ تعیین محدوده عمقی مناسب با امتحان کردن تمامی محدوده‌ها.

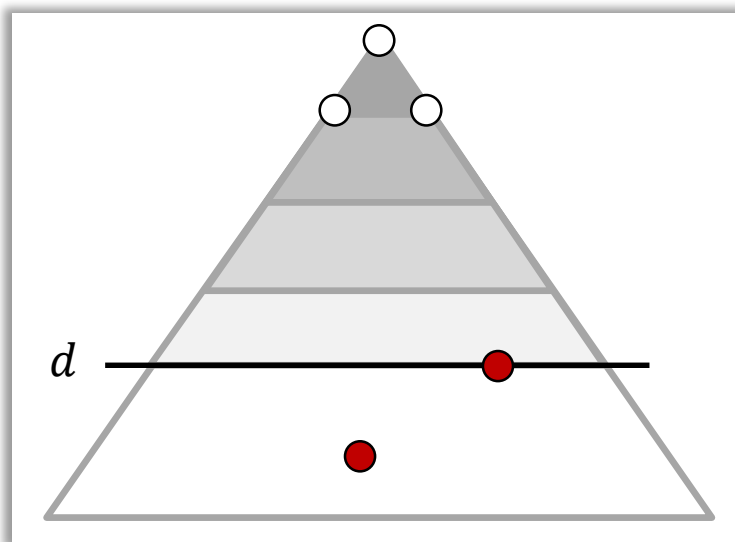


جستجوی عمیق کننده تکراری

۶۸

□ ایده. ترکیب مزایای جستجوی سطحی و جستجوی عمقی.

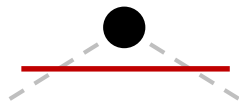
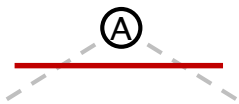
- یک جستجوی عمقی با محدوده عمقی ۰ انجام بده. اگر راه حل پیدا نشد ...
- یک جستجوی عمقی با محدوده عمقی ۱ انجام بده. اگر راه حل پیدا نشد ...
- یک جستجوی عمقی با محدوده عمقی ۲ انجام بده. اگر راه حل پیدا نشد ...
- یک جستجوی عمقی با محدوده عمقی ۳ انجام بده. اگر راه حل پیدا نشد ...
- ...



جستجوی عمیق کننده تکراری

۶۹

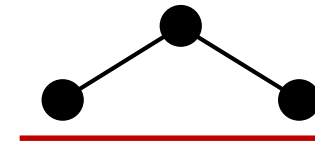
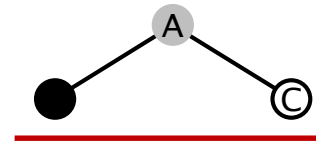
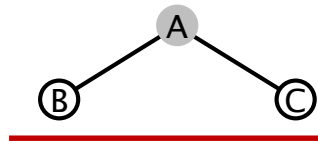
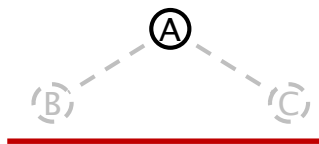
□ تکرار اول. $l = 0$



جستجوی عمیق کننده تکراری

۷۰

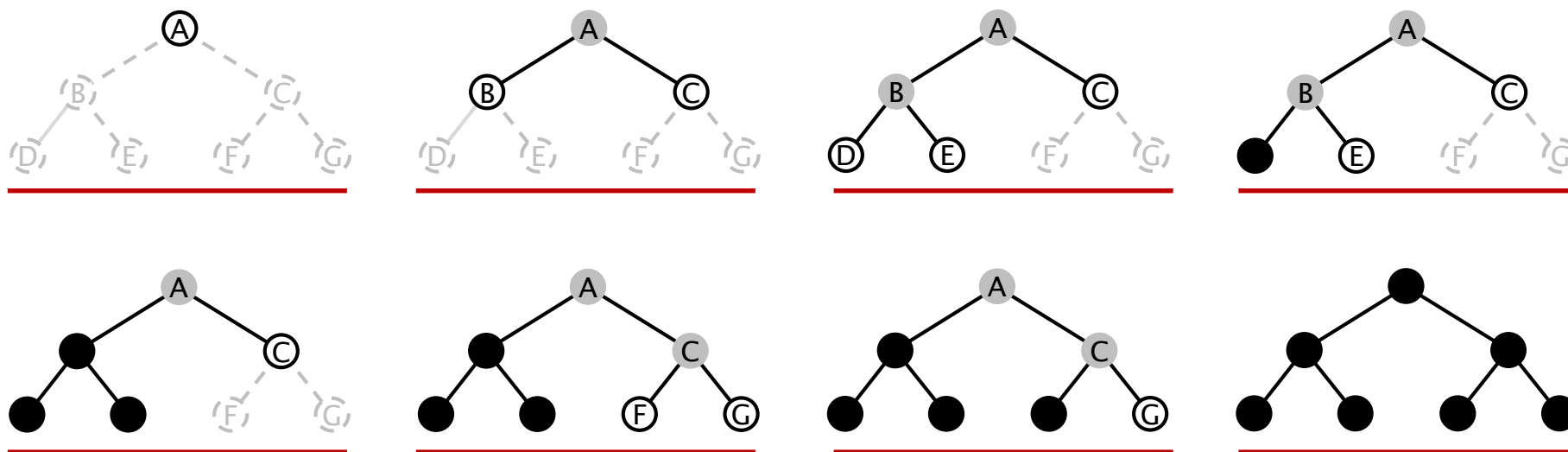
تکرار دوم. $l = 1$ □



جستجوی عمیق کننده تکراری

۷۱

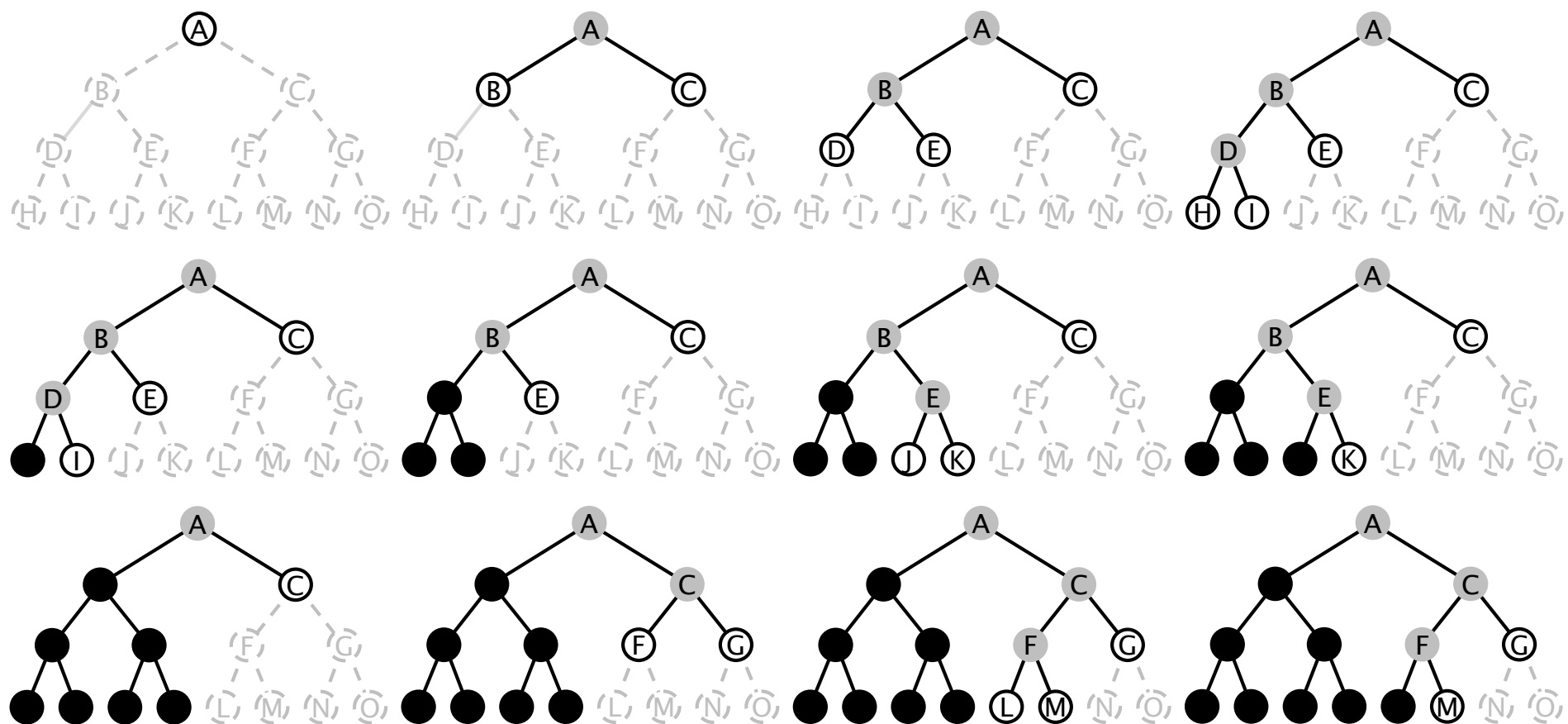
□ تکرار سوم. $l = 2$



جستجوی عمیق کننده تکراری

۷۲

تکرار چهارم. $l = 3$ □



ارزیابی جستجوی عمیق کننده تکراری

۷۳

ترکیب مزایای استراتژی‌های
سطحی و عمقی!

$$db^1 + (d-1)b^2 + (d-2)b^3 + \dots + 1b^d \in O(b^d)$$

$$O(bd)$$

□ کامل؟ بله

[مانند جستجوی سطحی]

□ بهینه؟ بله

[مانند جستجوی سطحی]

□ پیچیدگی زمانی؟ نمایی

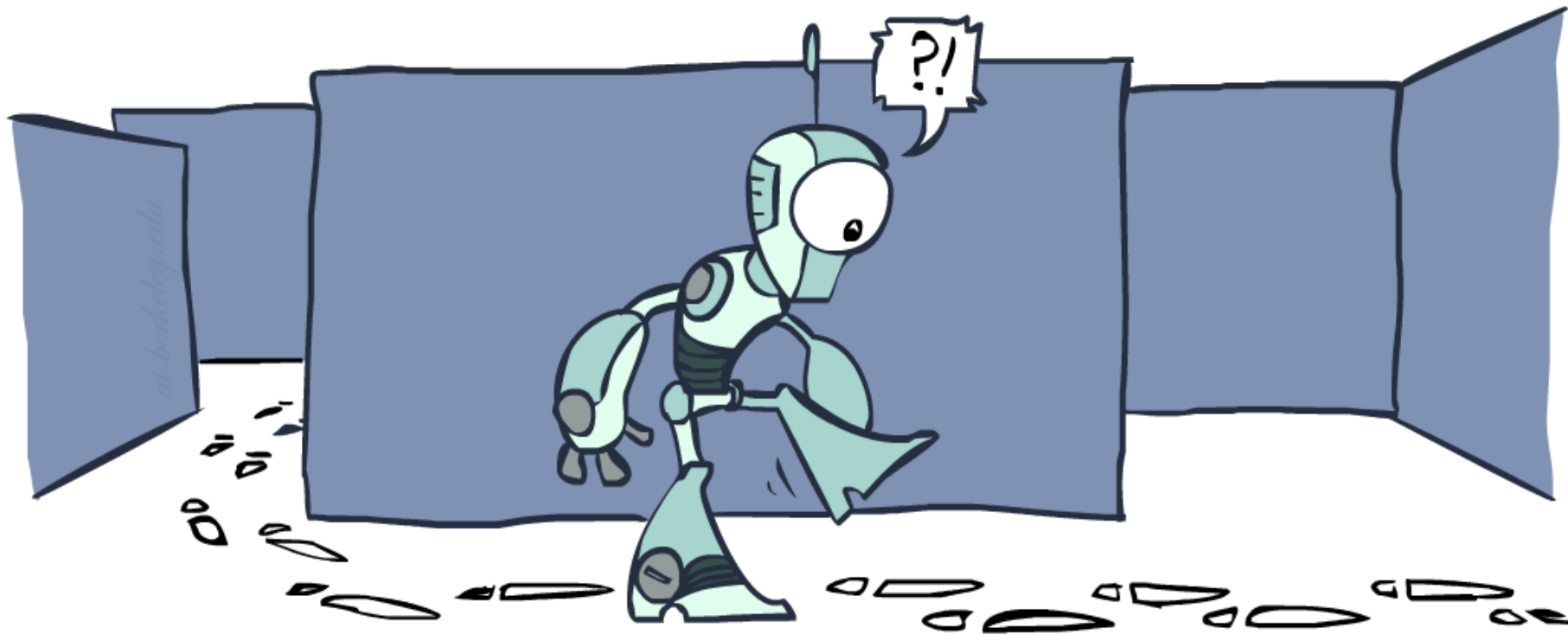
[مانند جستجوی عمقی]

□ پیچیدگی حافظه؟ خطی

[مانند جستجوی عمقی]

جستجوی گرافى

۷۴

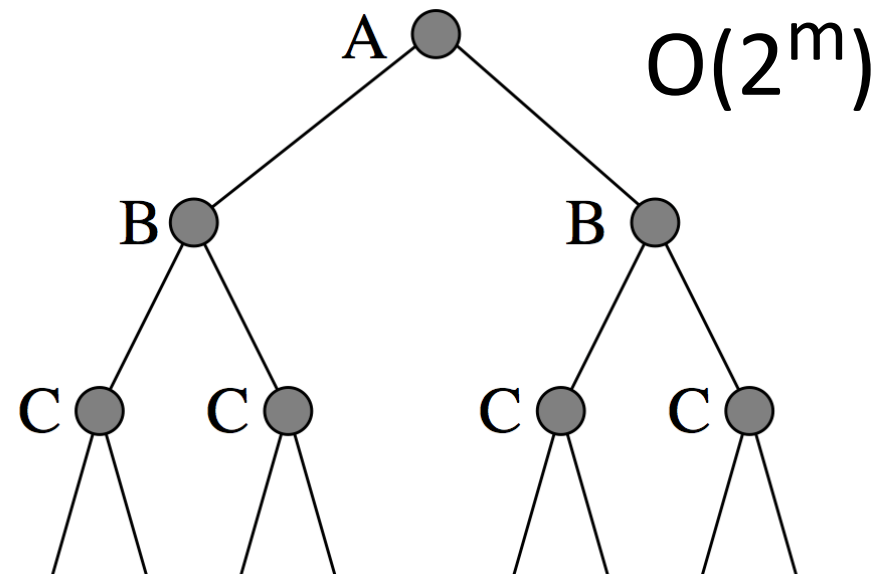


حالت‌های تکراری

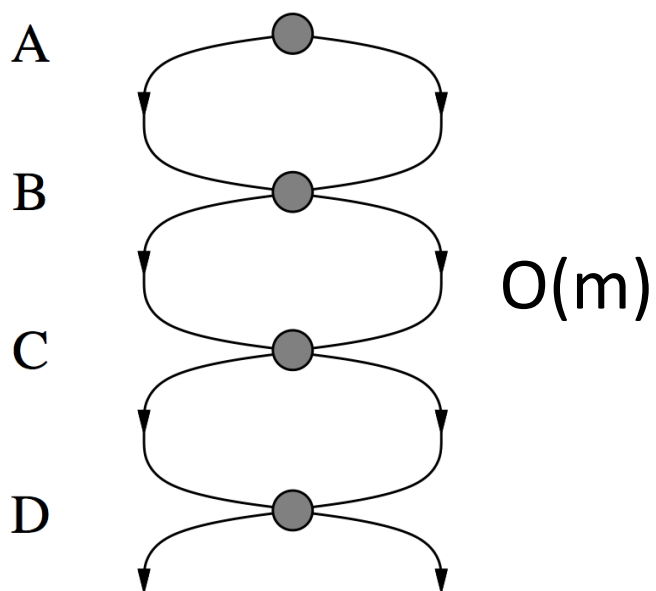
۷۵

□ شکست در تشخیص حالت‌های تکراری می‌تواند یک مسئله خطی را به یک مسئله نمایی تبدیل کند!

درخت جستجو



گراف فضای حالت



حالت‌های تکراری

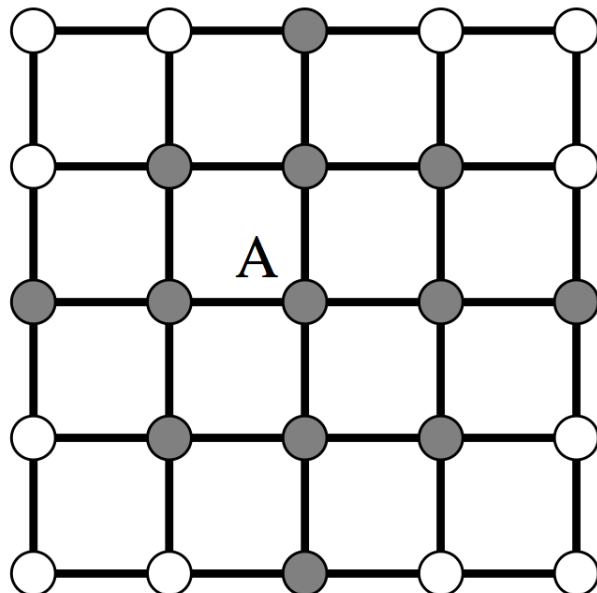
۷۶

□ شکست در تشخیص حالت‌های تکراری می‌تواند یک مسئله خطی را به یک مسئله نمایی تبدیل کند!

درخت جستجو

$$O(4^m)$$

گراف فضای حالت



$$O(2m^2)$$

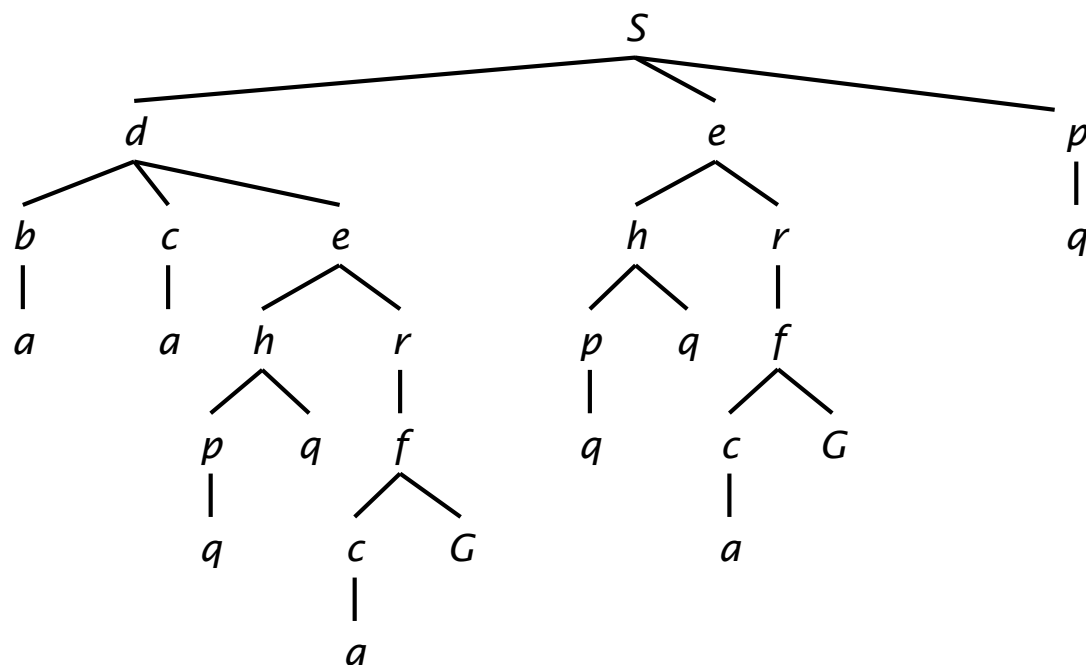
□ به ازای $m = 20$

□ تعداد حالت‌های گراف: ۸۰۰

□ تعداد گره‌های درخت: ۱۰۹۹۵۱۱۶۲۷۷۷۶

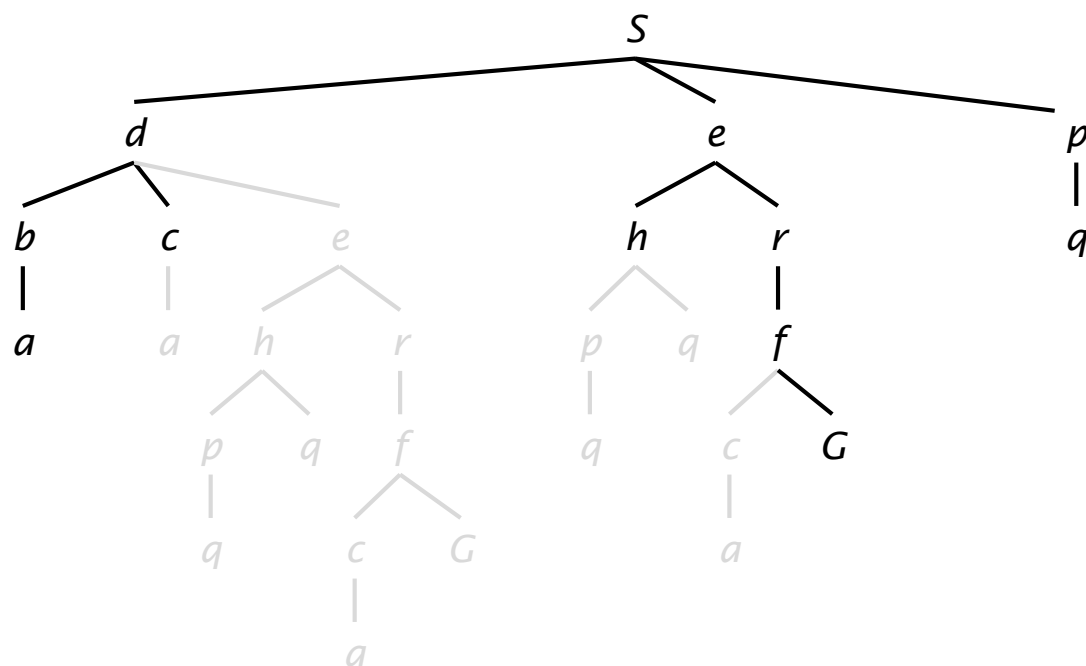
حالت‌های تکراری

□ به عنوان مثال، در جستجوی سطحی گسترش حالت‌هایی که قبلاً گسترش یافته‌اند، کاملاً بی‌فایده است.



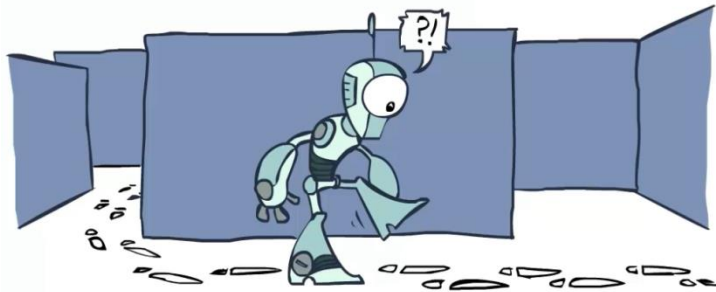
حالت‌های تکراری

□ به عنوان مثال، در جستجوی سطحی گسترش حالت‌هایی که قبلاً گسترش یافته‌اند، کاملاً بی‌فایده است.



جستجوی گرافی

۷۹



□ ایده. اجتناب از گسترش دادن حالت‌های تکراری.

□ پیاده‌سازی.

□ جستجوی درختی به همراه مجموعه گره‌های گسترش یافته

□ درخت جستجو را گره به گره گسترش بده، اما ...

□ قبل از گسترش دادن یک گره، بررسی کن که حالت آن گره یک حالت جدید باشد

□ در غیر این صورت گره بعدی را انتخاب کن

□ مهم. حالت گره‌های گسترش یافته در یک **مجموعه** ذخیره می‌شوند و نه در یک لیست.

□ س. آیا جستجوی گرافی ممکن است باعث ناکامل شدن یک استراتژی جستجو گردد؟

□ س. آیا جستجوی گرافی ممکن است باعث غیربهبود شدن یک استراتژی جستجو گردد؟

الگوریتم عمومی جستجوی گرافی

۸۰

```
function GRAPH-SEARCH(problem) returns a solution, or failure
  initialize the frontier using the initial state of problem
  initialize the explored set to be empty
  loop do
    if the frontier is empty then return failure
    choose a leaf node and remove it from the frontier
    if the node contains a goal state then return the corresponding solution
    add the node to the explored set
    expand the chosen node, adding the resulting nodes to the frontier
    only if not in the frontier or explored set
  end
```

انتخاب

آزمایش هدف

گسترش

□ ایده جستجوی گرافی.

□ ذخیره کردن گره‌های گسترش یافته.

□ جلوگیری از گسترش دادن گره‌هایی که قبلاً گسترش یافته‌اند.

جستجوی ناآگاهانه: خلاصه

□ فرموله سازی.

□ فرموله سازی مسئله اغلب نیاز به انتزاع جزییات مسئله دارد، تا بتوان فضای حالتی به دست آورد که به صورت مقرون به صرفه ای قابل کاوش کردن و جستجو باشد.

□ استراتژی های ناآگاهانه.

□ انواع مختلفی از استراتژی های ناآگاهانه بر اساس ترتیب انتخاب گره ها برای گسترش دادن وجود دارد.

□ استراتژی عمیق کننده تکراری.

□ مصرف حافظه جستجوی عمیق کننده تکراری دارای مرتبه خطی است و زمان خیلی بیشتری نسبت به سایر روش های ناآگاهانه مصرف نمی کند.

جستجو و مدل‌های انتزاعی از دنیای واقعی

۸۲

- الگوریتم‌های جستجو بر روی مدل‌های انتزاعی از دنیای واقعی جستجو می‌کنند.
- کیفیت جستجوی شما حداکثر به خوبی مدل شما از دنیای واقعی است.

